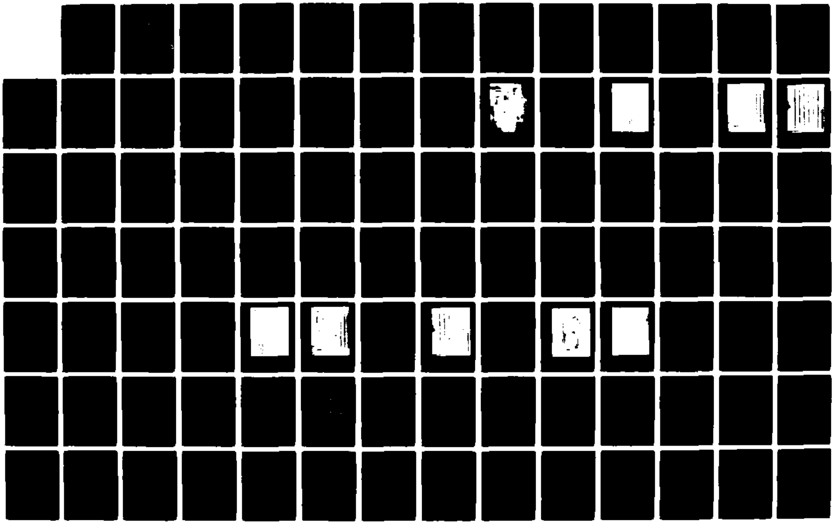
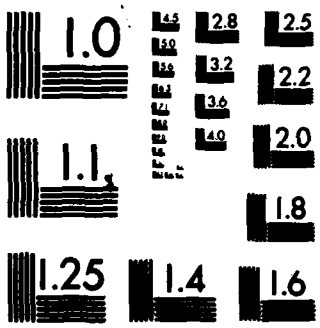
AD-A129 024 INTEGRATED ASSURANCE ASSESSMENT OF A RECONFIGURABLE DIGITAL FLIGHT CONTROL SYSTEM(U) LOCKHEED-GEORGIA CO MARIETTA W G NESS ET AL. APR 83 DOT/FAA/CT-82/154 1/2

UNCLASSIFIED NAS2-11179 F/G 1/3 NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

(12)

DOT/FAA/CT-82/154

Integrated Assurance Assessment of a Reconfigurable Digital Flight Control System

W. G. Ness
R. M. Davis
J. W. Benson
M. K. Smith
Lockheed-Georgia Company
Marietta, Georgia 30063

D. Eldredge
FAA Technical Center
Atlantic City Airport, New Jersey 08405

April 1983

Final Report

This document is available to the U.S. public through the National Technical Information Service, Springfield, Virginia 22161.

US Department of Transportation
Federal Aviation Administration
Technical Center
Atlantic City Airport, N.J. 08405

NOTICE

Technical Report Documentation Page

1. Report No. DOT/FAA/CT-82/154	2. Government Accession No. AD-A129 024	3. Recipient's Catalog No.

4. Title and Subtitle
INTEGRATED ASSURANCE ASSESSMENT OF A RECONFIGURABLE DIGITAL FLIGHT CONTROL SYSTEM

5. Report Date
April 1983

6. Performing Organization Code
182-340-100

7. Author(s)
W. G. Ness, R. M. Davis, J. W. Benson, M. K. Smith, and D. Eldredge

8. Performing Organization Report No.

9. Performing Organization Name and Address
Lockheed-Georgia Company
Marietta, GA 30063

FAA Technical Center
Atlantic City Airport, NJ 08405

10. Work Unit No. (TRAIS)

11. Contract or Grant No.
NAS2-11179

13. Type of Report and Period Covered
Final
Feb. - Oct. 1982

12. Sponsoring Agency Name and Address
U.S. Department of Transportation
Federal Aviation Administration
Technical Center
Atlantic City Airport, NJ 08405

14. Sponsoring Agency Code

15. Supplementary Notes
This contract was funded through interagency agreement NASA NMI 1052.151 to NASA Ames Research Center through task order DOT-FA77WAI-738, Modification Numbers 5 and 6 from the FAA Technical Center (ACT-340).

16. Abstract

FAA Advisory Circular AC 25.1309-1 provides guidance material for demonstrating compliance with the requirements of Part 25 of the Federal Aviation Regulations for "flight-essential" and "flight-critical" avionics systems. This advisory circular outlines the use of quantitative safety analyses which may include: a) Probability analysis; b) fault tree analysis; c) failure modes and effects analysis; and d) other comparable techniques for determining compliance with the requirements of FAR 25.1309(b).

The objective of this study was to explore and demonstrate the integrated application of reliability, failure effects and system simulator methods in establishing the airworthiness of a "flight-critical" digital flight control system (DFCS). The emphasis was on the mutual reinforcement of the methods in demonstrating the system safety.

1 times 10 to the -9th power

It was concluded from this study that: a) The integrated approach can be used for the validation of "flight-essential" and "flight-critical" digital systems; b) the quantitative assessment of reliability (system failure probability) can be accurately predicted at less than 1×10^{-9} by the use of both the fault tree analysis and the analytical reliability prediction analysis; c) fault tree analysis must be augmented by failure modes and effects analysis which must be used below the circuit card level because of the complexities of the lower level analysis; and d) system simulation (fault insertion) confirms the correct implementation of the fault detection and fault tolerance capabilities of the system under study.

17. Key Words
Integrated Assurance Assessment
Failure Modes and Effects Analysis
Fault Tree
Fault Insertion
Failure Rates
RDFCS Pallet
Digital Flight Control System

18. Distribution Statement
Document is available to the U.S. public through the National Technical Information Service, Springfield, Virginia 22161

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
UNCLASSIFIED	UNCLASSIFIED		

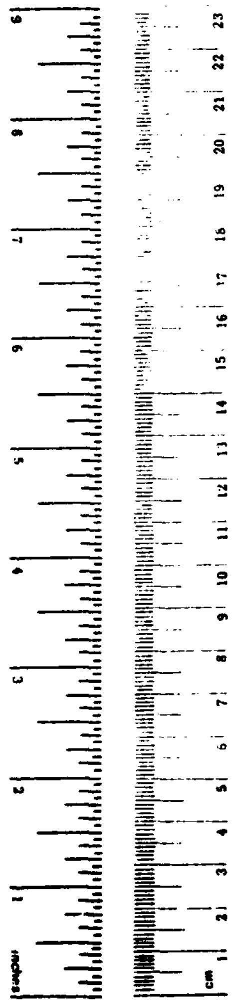
Form DOT F 1700.7 (8-72) Reproduction of completed page authorized

METRIC CONVERSION FACTORS

Approximate Conversions to Metric Measures

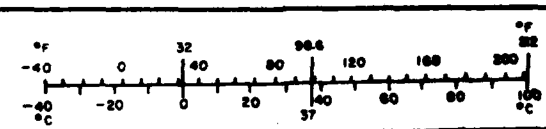
Symbol	When You Know	Multiply by	To Find	Symbol
		LENGTH		
in	inches	*2.5	centimeters	cm
ft	feet	30	centimeters	cm
yd	yards	0.9	meters	m
mi	miles	1.6	kilometers	km
		AREA		
in^2	square inches	6.5	square centimeters	cm^2
ft^2	square feet	0.09	square meters	m^2
yd^2	square yards	0.8	square meters	m^2
mi^2	square miles	2.6	square kilometers	km^2
	acres	0.4	hectares	ha
		MASS (weight)		
oz	ounces	28	grams	g
lb	pounds	0.45	kilograms	kg
	short tons (2000 lb)	0.9	tonnes	t
		VOLUME		
tsp	teaspoons	5	milliliters	ml
Tbsp	tablespoons	15	milliliters	ml
fl oz	fluid ounces	30	milliliters	ml
c	cups	0.24	liters	l
pt	pints	0.47	liters	l
qt	quarts	0.95	liters	l
gal	gallons	3.8	liters	l
ft^3	cubic feet	0.03	cubic meters	m^3
yd^3	cubic yards	0.76	cubic meters	m^3
		TEMPERATURE (exact)		
°F	Fahrenheit temperature	5/9 (after subtracting 32)	Celsius temperature	°C

*1 in = 2.54 (exactly). For other exact conversions and more detailed tables, see NBS Misc. Publ. 286, Units of Weights and Measures, Price $2.25, SD Catalog No. C13.10:286.



Approximate Conversions from Metric Measures

Symbol	When You Know	Multiply by	To Find	Symbol
		LENGTH		
mm	millimeters	0.04	inches	in
cm	centimeters	0.4	inches	in
m	meters	3.3	feet	ft
m	meters	1.1	yards	yd
km	kilometers	0.6	miles	mi
		AREA		
cm^2	square centimeters	0.16	square inches	in^2
m^2	square meters	1.2	square yards	yd^2
km^2	square kilometers	0.4	square miles	mi^2
ha	hectares (10,000 m^2)	2.5	acres	
		MASS (weight)		
g	grams	0.035	ounces	oz
kg	kilograms	2.2	pounds	lb
t	tonnes (1000 kg)	1.1	short tons	
		VOLUME		
ml	milliliters	0.03	fluid ounces	fl oz
l	liters	2.1	pints	pt
l	liters	1.06	quarts	qt
l	liters	0.26	gallons	gal
m^3	cubic meters	35	cubic feet	ft^3
m^3	cubic meters	1.3	cubic yards	yd^3
		TEMPERATURE (exact)		
°C	Celsius temperature	9/5 (then add 32)	Fahrenheit temperature	°F



FOREWORD

This report describes an assurance assessment of a representative contemporary digital flight control system stressing the use of various methods in a complementary manner. The work was performed between February 1, 1982, and September 30, 1982, under contract number NAS2-11179. The work was sponsored and directed by the Federal Aviation Administration Technical Center, with the contract administered through the National Aeronautics and Space Administration - Ames Research Center under interagency agreement NAS NMI 1052.51 (Task Order DOT-FAA-77WAI-738).

TABLE OF CONTENTS

Section	Title	Page
1	Introduction and Summary	1
2	Objectives and Scope	5
3	Contract Task Summary	6
4	RDFCS and Simulator Descriptions	8
5	Fault Tree Analysis	19
6	Failure Mode and Effect Analysis	41
7	Fault Insertion	49
8	Failure Rate Development	58
9	Reliability Prediction Using CARSRA	63
10	Conclusions	79
11	References	82
APPENDIX A	FMEA Results	A-1
APPENDIX B	Fault Simulation Results	B-1
APPENDIX C	Processor Schematic Diagrams	C-1

LIST OF FIGURES

Figure	Title	Page
1	RDFCS Dual-Dual Configuration	9
2	RDFCS Simulator	12
3	CAPS Test Adapter and Computer Breakout Panel	14
4	Servo Simulator Panel	16
5	Discrete Switch Panel	17
6	Fault Tree Top Level	21
7	Sensing Function	23
8	Normal Acceleration Sensing	24
9	NO DUAL Annunciation	27
10	Computing Function	29
11	Channel 1A Rudder Command	30
12	FCC Processor Control Card	31
13	FCC Processor Data Path Card	32
14	Yaw Autopilot Servo Command Warning	34
15	Servo Functions	37
16	No. 1 Yaw Autopilot Servo	38
17	Multiple Failures During Crucial Phase	39
18	One Output, Two Input Conditions	44
19	Two Output, One Input Conditions	45
20	Processor Block Diagram	46
21	CAPS Test Adapter and Computer Breakout Panel	50
22	Servo Simulator Panel	51
23	Discrete Switch Panel	53

LIST OF FIGURES (Cont'd)

Figure	Title	Page
24	FCC with Processor Card Extended	55
25	FCC Processor Data Path Card	56
26	Markov Model of a Dual Stage	65
27	Markov Model Coding for RDFCS Sensor Stages	71
28	CARSRA Input	73
C1	Control Card Schematic Diagram	C-2
C2	Data Path Card Schematic Diagram	C-5

LIST OF TABLES

Table	Title	Page
1	Assurance Method Functions	3
2	System Failure Probability	40
3	Pin-Level FMEA	A-2
4	Faults Simulated	B-2
5	FCC Control Card Failure Rate	60
6	Predicted FCC Card Failure Rates	61
7	Predicted Failure Rates for Major RDFCS Components	62

1. INTRODUCTION AND SUMMARY

Under the FAA Technical Center's Digital System Program (182-340-100), an integrated assurance assessment of a contemporary digital flight control system was performed. The assurance methods of fault tree analysis, automated reliability prediction, failure mode and effect analysis, and fault insertion were applied in a complementary way to address the need for a workable approach to confirming the airworthiness of a critical digital system. The resulting assessment satisfied the requirements of Advisory Circular 25.1309-1 (Ref. 1), and is consistent with the validation requirements of RTCA Document DO-178 (Ref. 2).

The digital system used in the analysis was the Redundant Digital Flight Control System (RDFCS) procured jointly by the FAA and NASA-Ames Research Center in 1979. The RDFCS facility is located at NASA-Ames as a central part of the Digital Flight Control Systems Verification Laboratory, a unique facility for research into the assurance issues of digital systems. Volume II of this report describes the RDFCS as it would be in a production configuration, including sensors and servos. The sensors and servos are not production-configuration equipment, and in fact, they are simulated in the RDFCS.

The assessment consisted of the following major tasks:

- Application of fault tree analysis, starting at the highest system functional level, proceeding to the hardware circuit card level, and to the module level for the processors.

- Development of a representative set of failure rates for the relevant hardware items.

- Application of an automated reliability prediction program, CARSRA, to the system failure modes affecting airworthiness.

- Application of failure mode and effect analysis to integrated circuit pin faults of three processor modules.

- Definition of faults to be inserted in the RDFCS to determine the effect of the fault when analysis was not feasible, and of other faults to confirm the manual analysis. These faults were subsequently inserted and the effects recorded.

Among the conclusions and observations resulting from this study are that:

- The integrated approach used here is capable, with diligent application, of establishing the airworthiness of a Digital Flight Control System (DFCS) within the context of AC 25.1309-1. Specifically, this approach addresses those system aspects shown in Table 1, including freedom from single-point failure modes and system failure probability.

- The integrated assurance approach used in this study should be considered for use in validating other digital systems, including DFCS, in compliance with AC 25.1309-1.

- The quantitative assessment of system failure probability by two methods (fault tree analysis and analytical reliability prediction) offers increased assurance that the system meets the quantitative requirements of AC 25.1309-1. For a flight-critical system, this requirement is that the system failure probability not exceed 1×10^{-9} per hour of flight for each critical function the system performs.

- Fault insertion confirms that the fault detection capability and the fault tolerance capability described in the system documentation are actually implemented in the system. Since the fault tree analysis is based largely on the system response to faults as described in the system documentation, the fault insertion confirms that the fault tree analysis correctly reflects the behavior of the actual system in the presence of faults.

- The fault tree analysis generates software test requirements in terms of functions which the software must perform. These, in turn, provide a check of function criticality and of test requirements generated in accordance with RTCA Document DO-178.

- Fault tree analysis proved unwieldy below the circuit card level, because at lower levels many more functions are being performed than there are hardware failure modes. Failure mode and effect analysis was accomplished successfully at the integrated circuit pin level.

- As a training facility and a Reconfigurable Test Bed, the RDFCS facility has significant and valuable capabilities for investigating assurance issues of currently definable DFCS architectures. It also has potential enhanced capability in certain areas, such as automated insertion of pin-level faults, for confirmation of analytically determined failure effects.

- The comparison of the time or cost required for the integrated approach reported here with that required for other possible assurance approaches was not specifically addressed in this study. However, the time required for the integrated approach is

TABLE 1. ASSURANCE METHOD FUNCTIONS

SYSTEM ASPECT	ASSURANCE METHOD: PRIMARY	ASSURANCE METHOD: CONFIRMATION
FAILURE EFFECTS		
- COMPONENT	FAULT TREE ANALYSIS	FAULT INSERTION
- DIGITAL MODULE	FAULT TREE ANALYSIS, FAILURE MODE AND EFFECT ANALYSIS	FAULT INSERTION
- DIGITAL INTEGRATED CIRCUIT	FAILURE MODE AND EFFECT ANALYSIS	FAULT INSERTION
- UNTRACTABLE CASES	FAULT INSERTION	
FAULT DETECTION/ ANNUNCIATION	FAULT TREE ANALYSIS	FAULT INSERTION
SOFTWARE FUNCTION IMPLEMENTATION	SOFTWARE TEST PROGRAM	FAULT TREE ANALYSIS
NO SINGLE-POINT FAILURE MODES	ABOVE, AS RELEVANT	ABOVE, AS RELEVANT
SYSTEM FAILURE PROBABILITY	RELIABILITY PREDICTION PROGRAM	FAULT TREE ANALYSIS QUANTITATIVE EVALUATION

expected to compare favorably with that for other approaches, assuming the same depth of analysis. The cost should also compare favorably, provided a facility suitable for fault insertion is available.

2. OBJECTIVES AND SCOPE

OBJECTIVES

The primary objective of this contract was to explore and demonstrate the integrated application of reliability, failure effects, and system simulator methods in establishing the airworthiness of a flight-critical digital flight control system. The emphasis was on the mutual reinforcement of the methods, with results oriented toward inclusion in an FAA Data Base.

SCOPE

The scope of the effort was primarily limited to assessment of the RDFCS in the automatic landing maneuver under Category IIIa conditions as defined in AC 120-28C (Ref. 3). Application of methods below the system level was on a selective basis and focused within the digital portions of the system. Installation-dependent effects, such as failure of RDFCS components induced by failure of components in other systems, were not considered.

3. CONTRACT TASK SUMMARY

SYSTEM DESCRIPTION

A baseline configuration of the RDFCS shall be defined, and a corresponding analytical description shall be prepared as necessary to perform the integrated assessment. This description may include existing documentation for the RDFCS, and as necessary, it shall include additional components (e.g., secondary flight control) needed to reflect a realistic DFCS.

FAULT TREE ANALYSIS

A fault tree analysis beginning at the system level is required. The analysis shall be extended the integrated circuit pin level for at least three digital modules.

FAILURE RATES

A set of representative failure rates for the components and parts of the RDFCS shall be developed as necessary to evaluate the fault tree for failure probability.

FAULT SIMULATION CASES

A number of simulated fault conditions shall be defined for insertion in the RDFCS simulator. These faults shall be for two purposes: to confirm the assumptions underlying the fault tree analysis, and to resolve uncertainty of the effect of the fault when analysis is not tractable.

FLIGHT CASE TRANSITIONS

A go-around flight case shall be installed on the RDFCS simulator, and transition capability shall be installed to transition the airplane from approach to landing and landing to go-around flight cases.

CARSRA RELIABILITY PROGRAM

The CARSRA reliability program shall be applied to the RDFCS. The application shall be made in such a way as to be instructive for future applications of CARSRA to other system.

4. RDFCS AND SIMULATOR DESCRIPTIONS

RDFCS

The RDFCS is described in considerable detail in Volume II of this report. The description presented here summarizes the system architecture. In most operational modes, the system is fail passive, with a dual channel configuration. For automatic landings under Category IIIa conditions, the system can be brought into a dual-dual fail-operational, fail-passive configuration. The classification dual-dual relates primarily to the four computer channels in the system. Each of the two flight control computers (FCC) has two channels which run frame-synchronously, with each channel driving one coil of a dual-coil servo in each axis. Any indication of disagreement between the two channels in an FCC causes the servo connected to that FCC to be disengaged by removing hydraulic pressure. Figure 1 summarizes the dual-dual configuration.

Monitoring Configuration and Implementations

Extensive monitoring is employed in the RDFCS for fault detection. Coil current comparators for each servo provide coverage of faults resulting in erroneous commands to the servo coils. They also provide coverage for broken wire faults between the FCC and the servo or failures of the coils themselves. These monitors, which are described in Volume II, Sections 5.1.1.6.2 through 5.1.1.6.5, are made more effective by the insertion of opposing 5 ma bias currents. The bias currents permit circuit integrity to be monitored even when the FCC is not commanding the servo to a new position, such as when the aircraft is flying through very calm air at a stable attitude. It may be noted that this type of monitoring is equally applicable to analog and digital systems.

Response of the autopilot servos to commands from the servo amplifiers is monitored by modulator piston position signals fed back to the FCC (Vol. II, Sections 5.1.1.6.3 through 5.1.1.6.5). The feedback signals are averaged and passed through a high-pass filter to get a modulator rate that is compared with coil current. This comparison is used to detect jamming

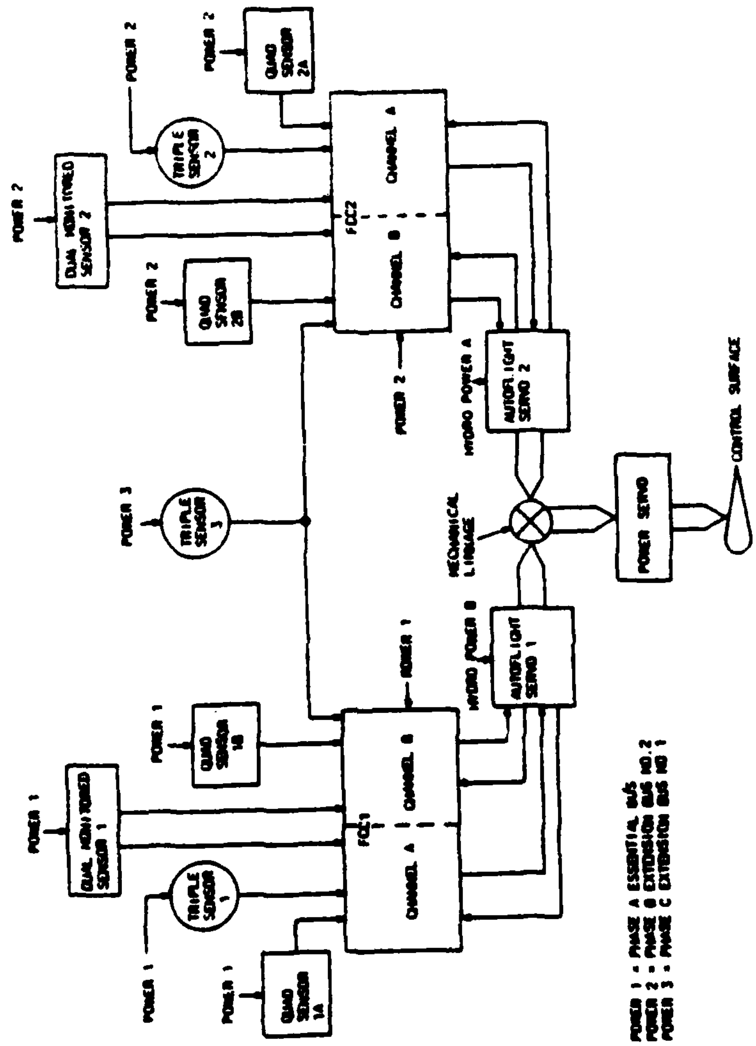


Figure 1. RDFCS Dual-Dual Configuration

of the modulator piston, runaway conditions, or loss of hydraulic power. This type of monitoring also can be applied to either analog or digital systems.

In the pitch-axis servos, modulator piston position monitoring is implemented in hardware. In the other two axes, it is implemented in software. Together, the coil current monitoring and modulator piston monitoring detect any servo fault which prevents the servo from responding to commands. They also detect any fault in a computer channel which prevents that channel from generating a reasonable command for the servos in each of the three axes. All monitors and feedback sensors are dual to increase reliability.

Each computer channel has an iteration monitor implemented in hardware (Vol. II, Figures 5.1.2.1.2 through 5.1.2.1.3). This monitor observes the state of a discrete software variable which is changed at the end of each iteration of the foreground software. Since this software executes at a 20 Hz rate, the result is a 10 Hz square wave. Should the processor short-loop or hang up, the 10 Hz wave will not be presented and the iteration monitor will withdraw its input to the engage logic and the FCC will disengage.

Sensor monitoring is primarily accomplished by comparison and by validity discretes generated by the sensors (Vol. II, Sec. 5.1.2.4 through 5.1.2.8). There is no one place that sensor monitoring takes place, since all four computer channels incorporate the monitoring function. This ensures that the circuitry involved in getting the sensor signals to each channel is included in the monitoring.

The gyro and accelerometer discretes are generated as described in Volume II, Sections 5.11 through 5.12. The accelerometers are tested as described in Section 5.11 each time the system is powered up with the airplane on the ground.

The ILS receivers are checked using the square wave test of Volume II, Section 5.1.2.3.1.1.5. This test checks for failure of the localizer and glideslope beam deviation inputs. During landing, the outputs of both receivers are compared, with reliance on the self-monitoring to identify which receiver is bad if the signals disagree. The comparison monitoring is used to check wire integrity between the receiver and the computer channels. The other dual sensors are comparison monitored in the same way.

Even though each channel monitors sensors individually, any channel can initiate the NO DUAL annunciation, which is the primary indication that the system is not fail-operational. If any channel detects a second failure of a sensor type, it will cause its FCC to disengage, but the other FCC will remain engaged.

Although NO DUAL is the primary warning of loss of one sensor, NO ALIGN will be annunciated if the course signals from the two compass systems do not agree.

Other monitoring within the FCC involves comparison of active operating modes. If the two channels within an FCC disagree on which modes are engaged, and the disagreement lasts for more than 0.1 sec, the FCC will disengage. If the two FCC's disagree, SPLIT will be displayed on the Warning Annunciator Indicators. This monitoring, together with the sensor data transfers, will detect most faults of the cross-channel data transfer circuitry.

SIMULATOR DESCRIPTION

The RDFCS simulator is comprised primarily of the RDFCS pallet, shown in Figure 2, and a PDP 11/60 computer. The RDFCS pallet includes the Flight Control Computers (FCC), core memory, Modular Digital Interface Control Unit (MDICU), Servo Simulator Panel (SSP), Discrete Switch Panel (DSP), CAPS Test Adapters (CTA), and Computer Breakout Panels. The functions of these items are described in the remainder of this section.

PDP 11/60 Computer/Airplane Model

The PDP 11/60 computer hosts a discrete-state model of the airplane in which the RDFCS is installed. This airplane is a representative wide-body transport, and the model coefficients are changed according to flight case being simulated. Each flight case, then, is a point simulation of the airplane in a particular configuration and operating in a specific portion of the flight envelope. The airplane model executes at a 50 Hz rate.

As part of this study, a go-around case was added to the library of cases available. These cases are described and discussed in Reference 4. The go-around case is characterized as follows:

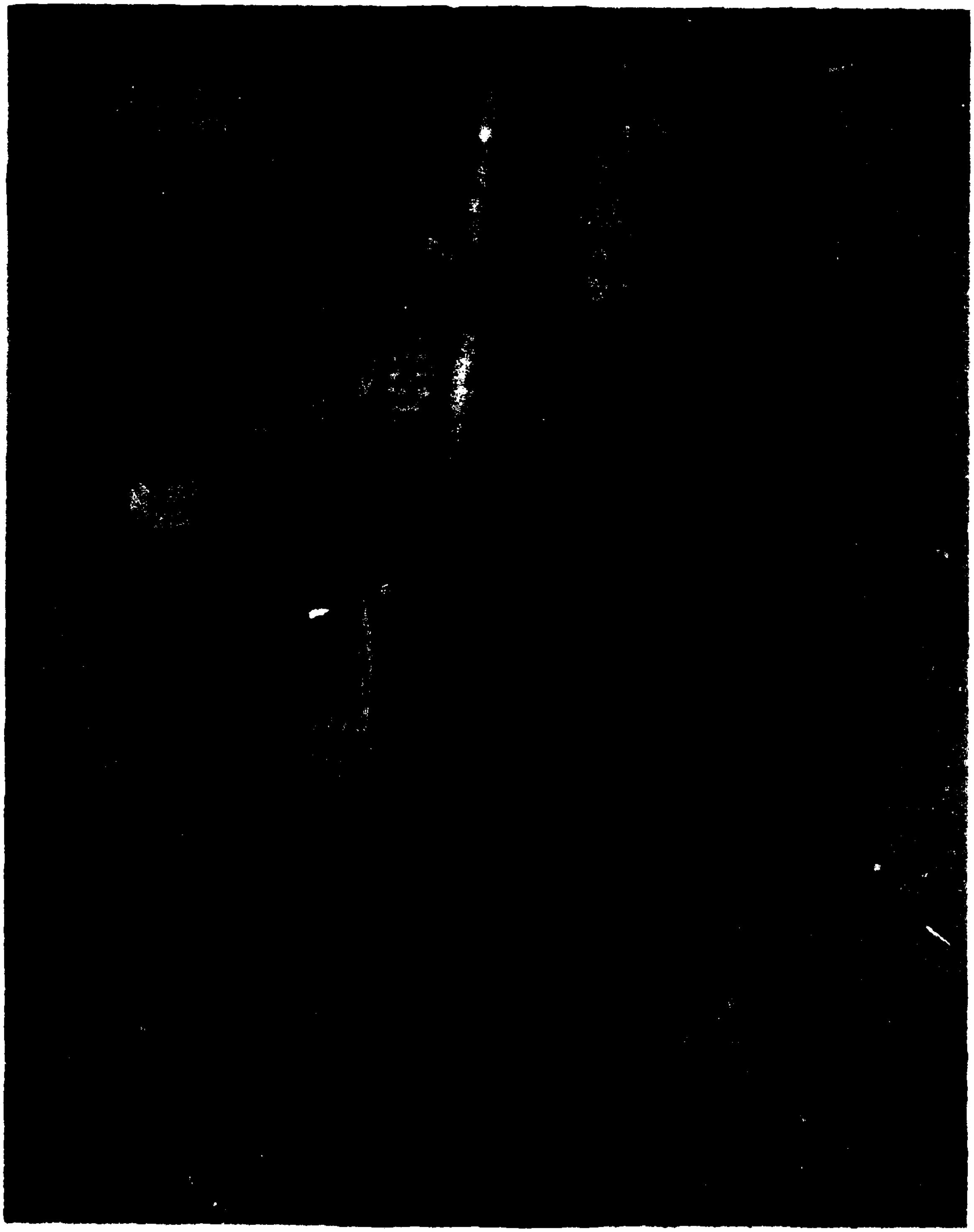


Figure 2. RDFCS Simulator

Airplane Weight	314,500 lb
Altitude	35 ft
Angle of Attack	10.91°
Indicated Air speed	168 kts
Flap Deployment	22°
Center of Gravity	25% of $\bar{c}$

Transition capability was added to go from approach conditions to landing conditions, and from landing to the new go-around case. The transitions involve changing the model coefficients and establishing new trim values. The transition capability has been installed and checked out successfully.

Modular Digital Interface Control Unit

The Modular Digital Interface Control Unit (MDICU) receives the output of the airplane discrete-state model through a communication link with the PDP 11/60 computer. The MDICU converts the various pieces of information into the form needed by the FCC's. For example, roll angle and pitch angle are converted to three-wire AC signals, properly scaled, while localizer deviation is coded in ARINC serial digital format. The MDICU is described more fully in Reference 5.

The MDICU incorporates provisions for the signal for the No. 1 sensor of each type to be ramped up or down. This facility is accessed by means of the HP 2645A terminal physically located in the pallet.

Computer Breakout Panels

Each sensor signal going from the MDICU to the FCC's can be interrupted at the Computer Breakout Panels by removing the appropriate jumper plug. Every FCC back connector pin is routed through one of these plugs. The lower portion of Figure 3 shows the rows of plugs for connector P1 and the "A" half of connector P2. Each FCC has its own breakout panel.

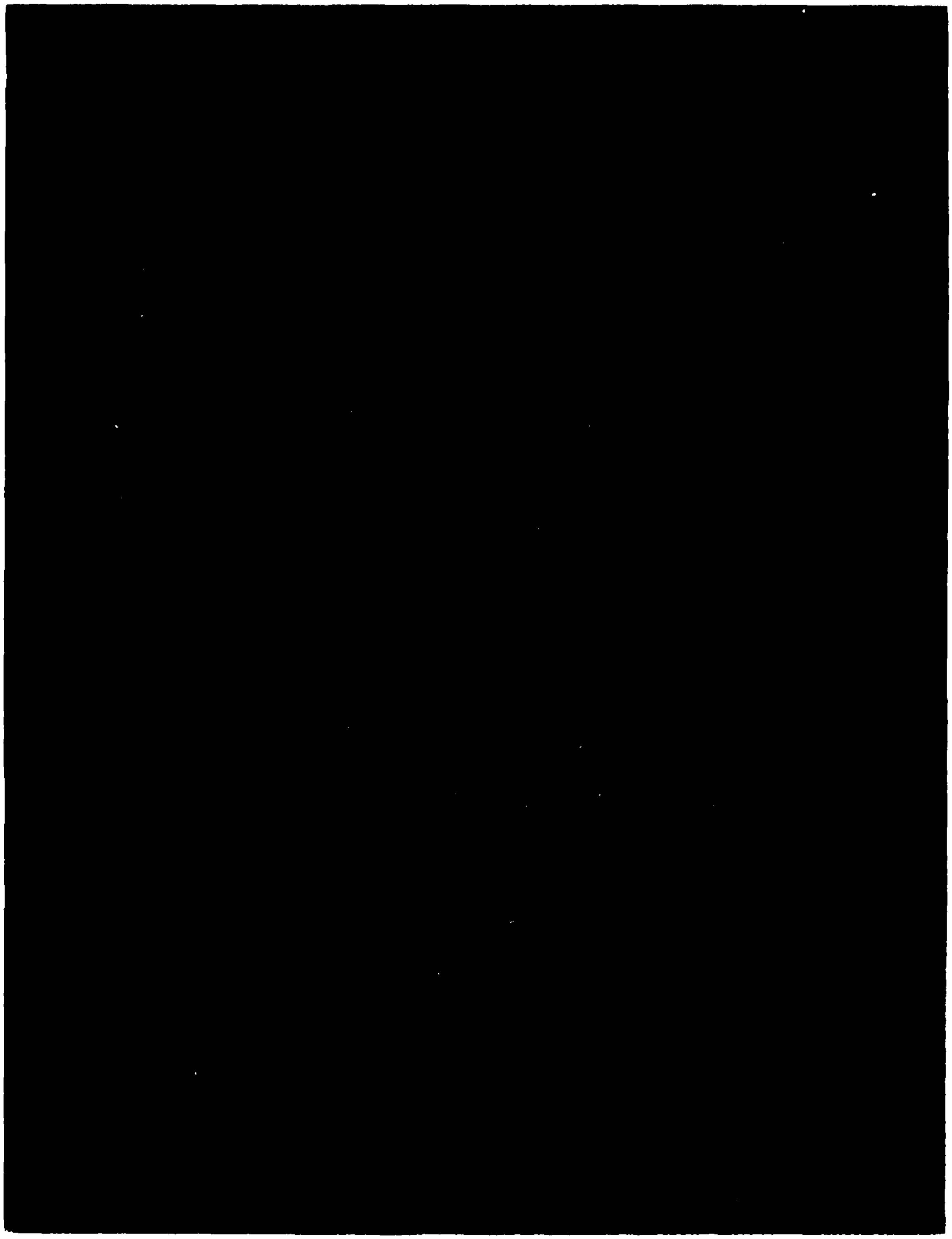


Figure 3. CAPS Test Adapter and Computer Breakout Panel

CAPS Test Adapters

Figure 3 also shows the CAPS Test Adapter (CTA) for one of the FCC's. The upper half of the CTA includes, on the right-hand side, four address and four data windows. An address can be loaded in each address window, and the corresponding data window used to display the data on the FCC A-side processor bus data lines every time the address appears on the address lines. The CTA also has other capabilities, such as providing a history of the last 16 bus transfers and changing the contents of a specific memory location within the FCC, but during the study only the address monitoring was used. Discrete variables representing sensor voter status were monitored visually via the data windows. Continuous variables, such as inputs to the servo amplifiers, were monitored by using the analog output posts below the appropriate data window to drive a strip-chart recorder.

The lower half of the CTA performs the same functions as the upper half, but for the B side of the FCC.

Servo Simulator Panel

The servo amplifier outputs from the FCC's are routed to the Servo Simulator Panel (SSP), shown in Figure 4. The SSP simulates the dynamics of the autopilot and power servos, and generates the required feedback signals such as modulator piston position. The SSP has circuits which can simulate a hardover or slowover command to a servo coil. It can also simulate a hardover or slowover of a modulator piston, including the modulator piston position feedback signal and the command to the power servo. All of these apply to the No. 1 servo of each type.

Discrete Switch Panel

The Discrete Switch Panel (DSP), Figure 5, is located just below the SSP. This panel provides a centralized location for switches such as hydraulic pressure switches and autopilot disconnect switches. The panel also includes switches that can be used to insert sensor validity faults. These faults can also be inserted by pulling the appropriate jumper plug on the FCC Breakout Panel.

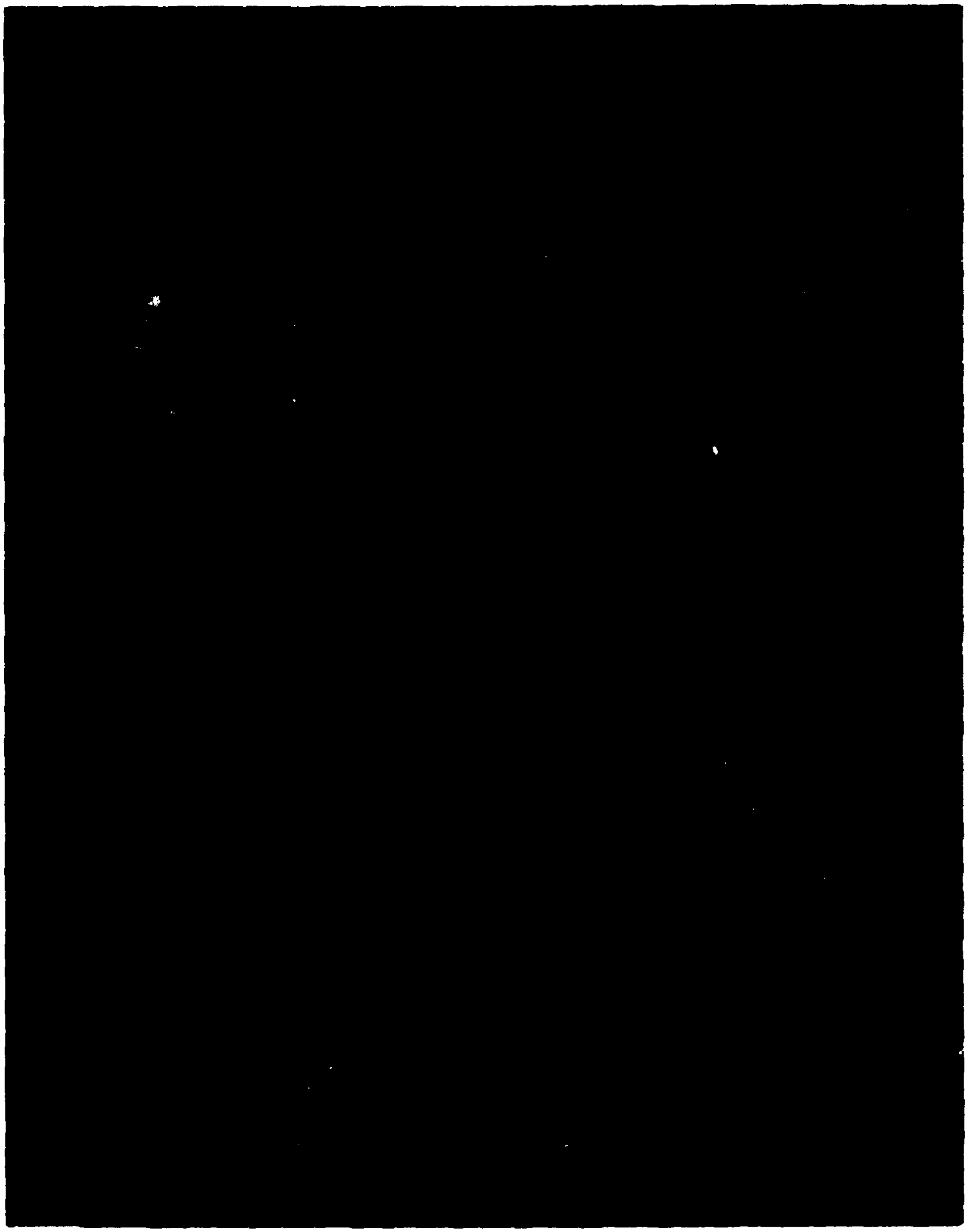


Figure 4. Servo Simulator Panel

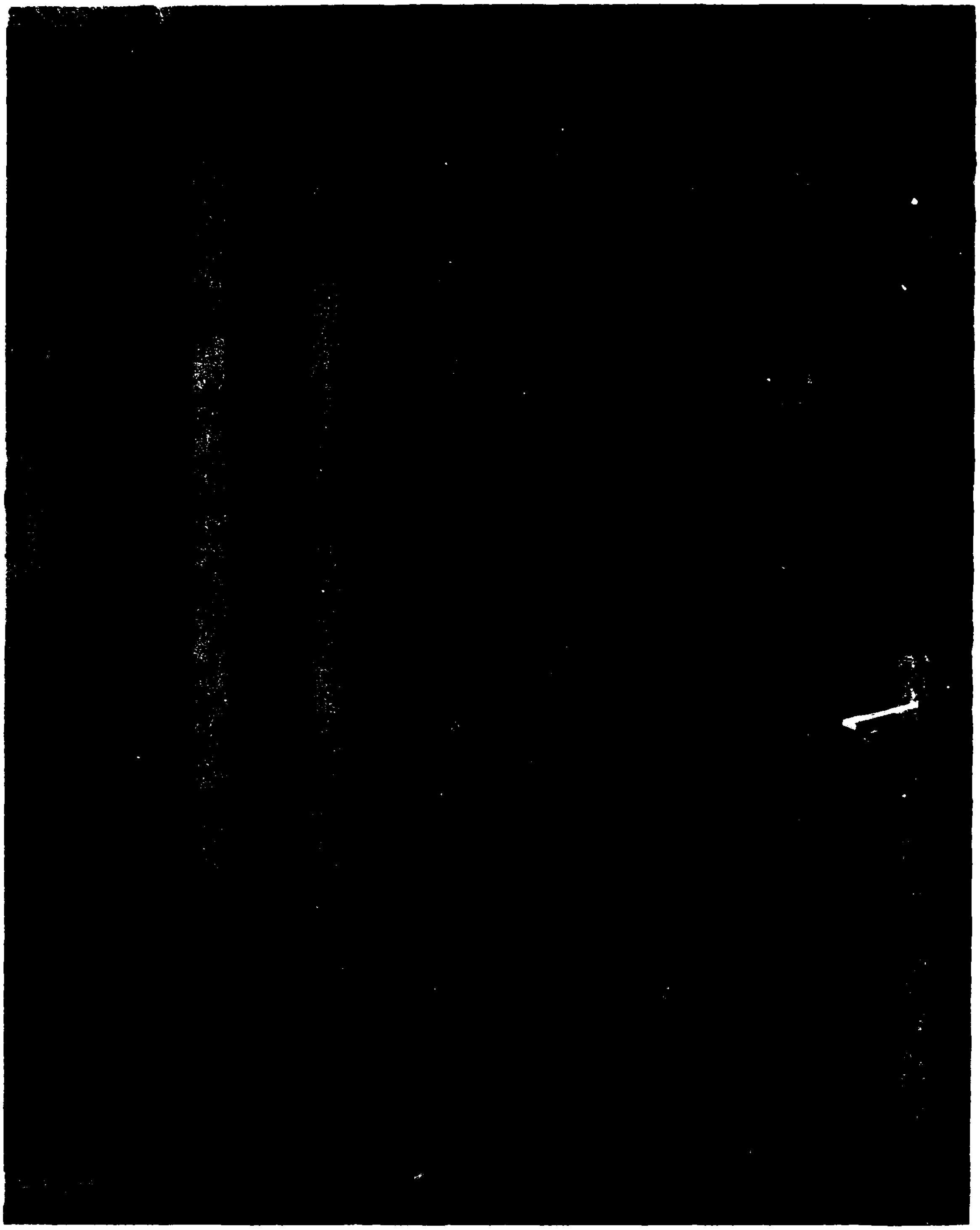


Figure 5. Discrete Switch Panel

Core Memory

The pallet also contains core memory for the FCC's. This is used for both data and program memory to provide flexibility and convenience in using the pallet to simulate other airplanes or DFCS architectures. As used in an airplane, the FCC's have the flight software stored in programmable read-only memory (PROM) and use random access memory (RAM) chips for data memory.

Glare-Shield Panel

The pallet also has a glare-shield panel, which is the control panel for the system as installed in an airplane. It includes the engage (bat handle) switches, mode select switches, altitude select knob, and other controls. The pallet also has a single ADI, HSI, radio altitude display, Mode Indicator, and Warning Annunciator Indicator.

5. FAULT TREE ANALYSIS

FAULT TREE ROLE IN INTEGRATED ASSURANCE

The integrated assurance assessment of the RDFCS begins with a fault tree analysis of the system function. Referring back to Table 1, the fault tree analysis has several functions. The first function is to assure that no system component has any failure mode which can result in system failure. Most of the components, such as the sensors and servos, have only a few failure modes which can be observed at the interfaces with the rest of the system. For these components, the fault tree analysis provides assurance that no failure modes can cause system failure. The assurance is obtained by reviewing the completed tree and determining that system failure can only occur as a result of multiple failures.

In general, digital modules (and therefore digital components) can have a substantial number of different failure modes. In such cases, it becomes quite laborious to continue the fault tree development to a level of detail sufficient to confirm that none of those failure modes can cause system failure. The second function of fault tree analysis is to identify which digital modules are involved in performing critical functions. The task of assuring that no single module level failure can cause system failure is performed with failure mode and effect analysis (FMEA).

A major benefit of fault tree analysis is that it focuses on the functions performed by the system elements, including those system elements involved in detecting faults and providing appropriate annunciation to the flight crew. Consequently, the third function of fault tree analysis is to confirm the adequacy of monitoring (i.e., fault detection and annunciation) in the system.

Fault tree analylsis is also used to identify specific software functions required for system operation, including fault monitoring implemented in software. The software test requirements for these functions are then specifically reviewed to confirm that these requirements are adequate. This fourth function of fault trees is discussed more fully and illustrated subsequently as the tree for the RDFCS is developed.

The fifth function of fault tree analysis is to provide an alternate means of computing the probability of system failure. This provides a check of the probability obtained from the CARSRA program to ensure that the CARSRA input does not have errors which would produce a false low probability of system failure.

FAULT TREE DEVELOPMENT

The fault tree analysis is based on the undesired event that the airplane has an unacceptable deviation from the desired flight profile during the last 150 feet of descent while executing an automatic landing, as shown in Figure 6. This portion of flight, which is the only flight phase during which the RDFCS performs a critical function, is termed the "crucial flight phase" in this report. Category IIIa conditions are assumed, so that the human pilot cannot complete the landing using visual cues should the RDFCS fail.

The analysis begins with the RDFCS in the dual-dual configuration. It should be noted that this configuration is available only after the Instrument Landing System (ILS) push-button has been used to select the Approach/Land (A/L) mode (Ref. Vol. II, Section 4.3.6.1). After this switch has been momentarily depressed, the A/L mode is transmitted to the FCC's and latched in. The switch is no longer needed, and therefore does not enter into the analysis.

The top event of Figure 6 can be caused by any of three conditions, or subevents. For convenience, these can be referred to as Level-2 events, with the top event considered to be at Level 1. The Level-2 events are shown as the middle row in Figure 6. The first of these is that the system design is in some manner deficient for the environmental conditions encountered. This includes the possibility that the conditions encountered are outside of the system design requirements; it also includes the possibility that the control laws are deficient for some conditions which may be expected. This possibility is outside the scope of this project and is not pursued here. References 6 and 7 address this subject. In particular, Section 3.3.1.3 of Reference 6 discusses establishing an upper bound on the probability of a deficient control law by statistical methods.

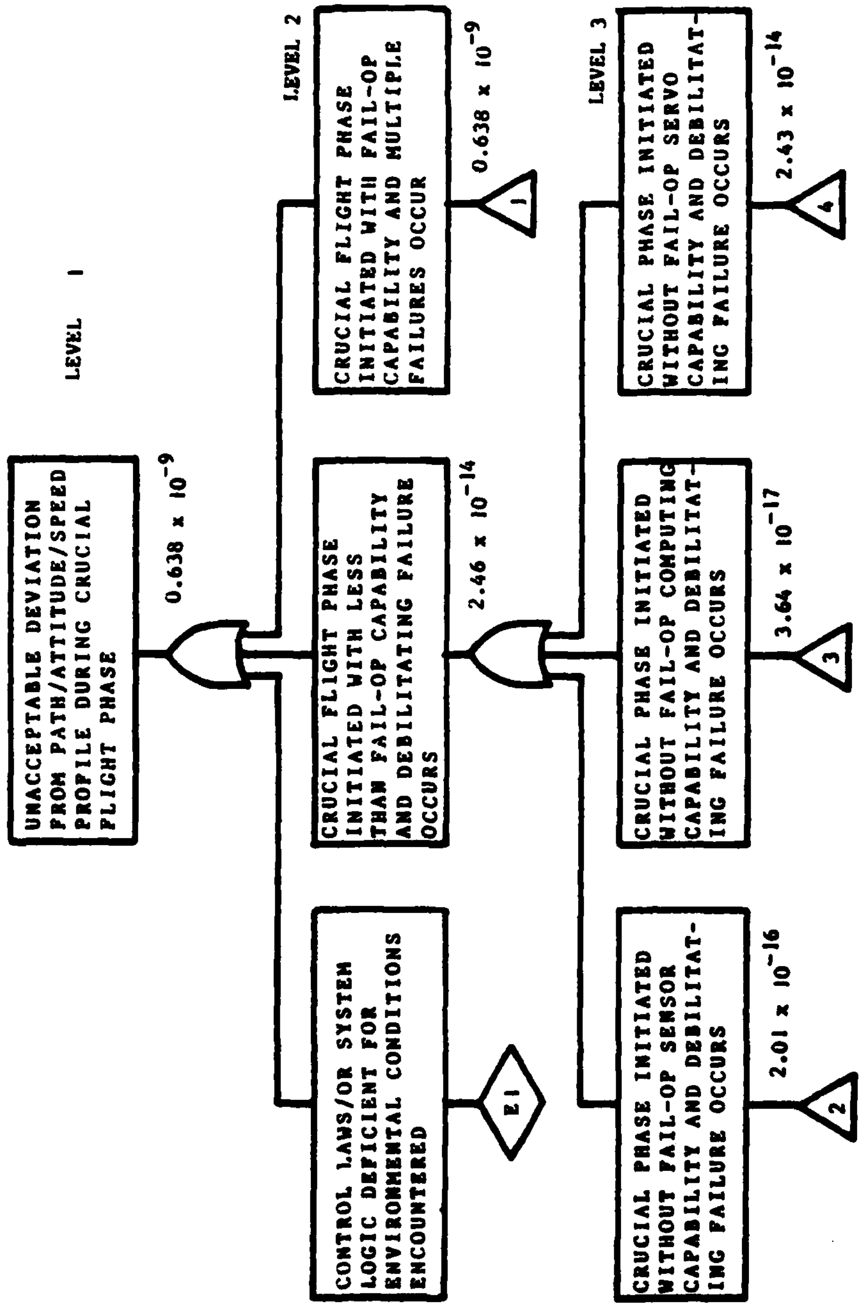


Figure 6. Fault Tree Top Level

The second of the Level-2 events occurs if the airplane enters the crucial phase with the RDFCS not fail-operational, and then a component failure occurs which prevents the system from completing the landing.

The third of the Level-2 events is that the crucial phase is entered with a fail-operational RDFCS, but multiple component failures occur before the end of the phase, and these failures result in RDFCS system failure.

The second of the Level-2 events, that the crucial phase is initiated without fail-operational capability, is expanded into three relevant functional areas, or Level-3 events: sensing aircraft attitude and position, computation of required outputs, and servo response to computed commands. The first of these, the sensing function, is expanded in Figure 7 into the various parameters needed by the FCC's in the automatic landing control laws. At this and higher levels, the fault tree is functionally oriented: failures are in terms of loss of function rather than loss of hardware.

The fault tree stub of Figure 8 extends the sensing function for normal acceleration to the individual hardware elements used to measure the acceleration and transmit it to the computers. The failure of the normal acceleration signal No. 1 to be present in all computer channels can be caused by loss of the sensor itself, associated wiring, or one of the circuit cards involved in receiving the signal and transmitting it to all channels. Volume II, Figure 5.1.1.3.1 shows the functional flow of these cards. The A24 Autoland Sensor Input and A27 Discrete Input Cards are both involved: The A24 card handles the analog acceleration signal and the A27 card handles the validity discrete signal. The processor itself is not involved in the data acquisition process and so is not shown. At this level, the transition has been made from required funcitons to the hardware which performs those functions.

Failure of the system to provide a NO DUAL annunciation is shown in Figure 9. This figure is of particular interest because of the explicit software function identified. A failure rate of zero is assigned to failure of this function, because it can be explicitly and exhaustively tested. Once it has been so tested, the probability of both NO DUAL annunciations failing because of a generic software error is taken to be zero. A generic software error is a discrepancy in the software which will

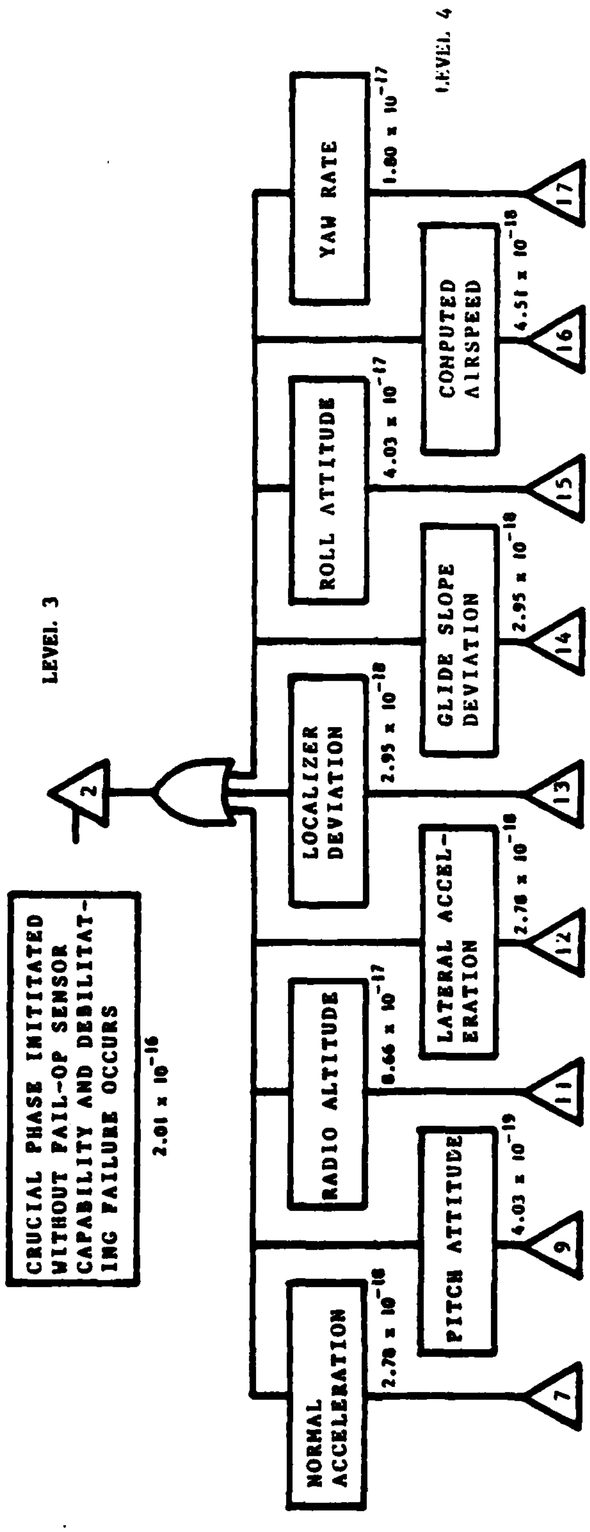


Figure 7. Sensing Function

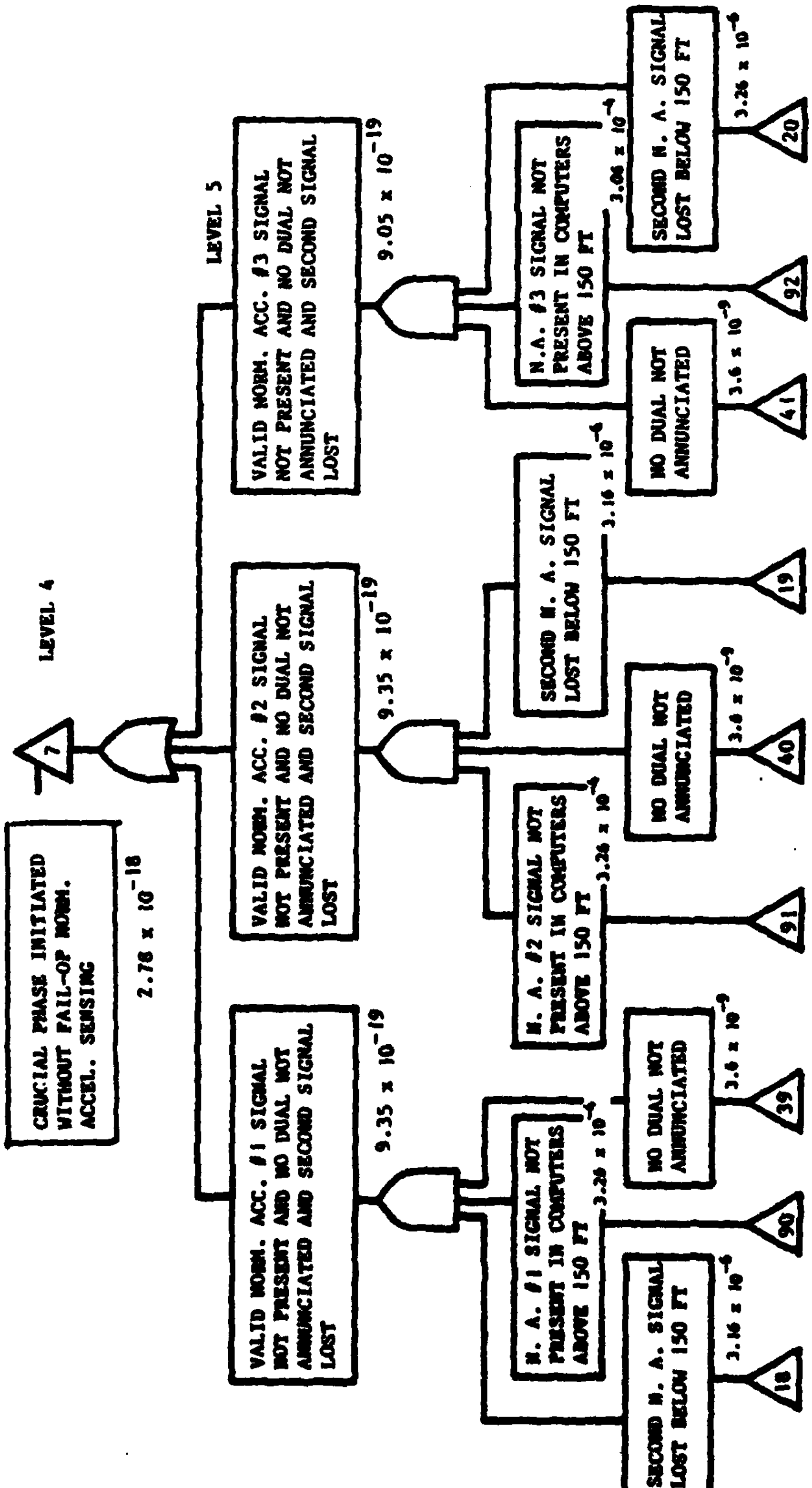
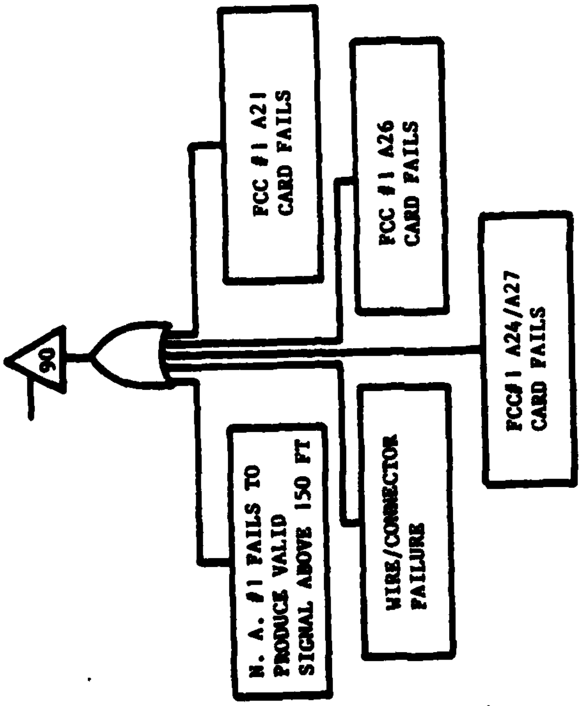


Figure 8. Normal Acceleration Sensing



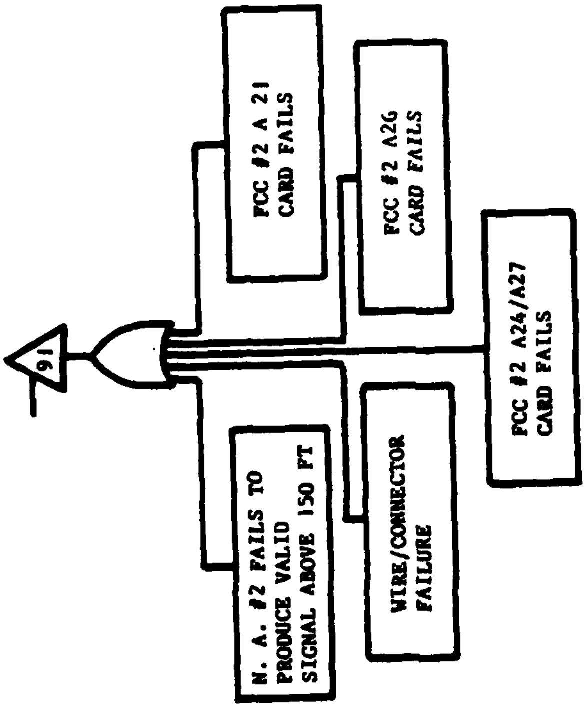


Figure 8. Normal Acceleration Sensing (Cont'd.)

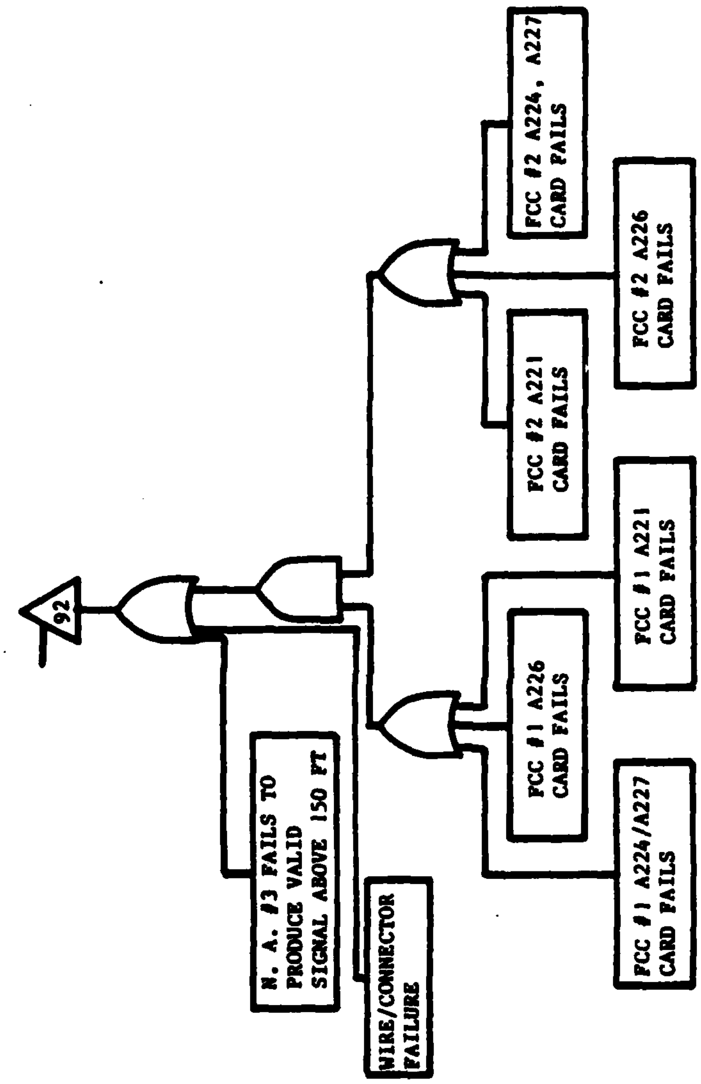


Figure 8. Normal Acceleration Sensing (Cont'd.)

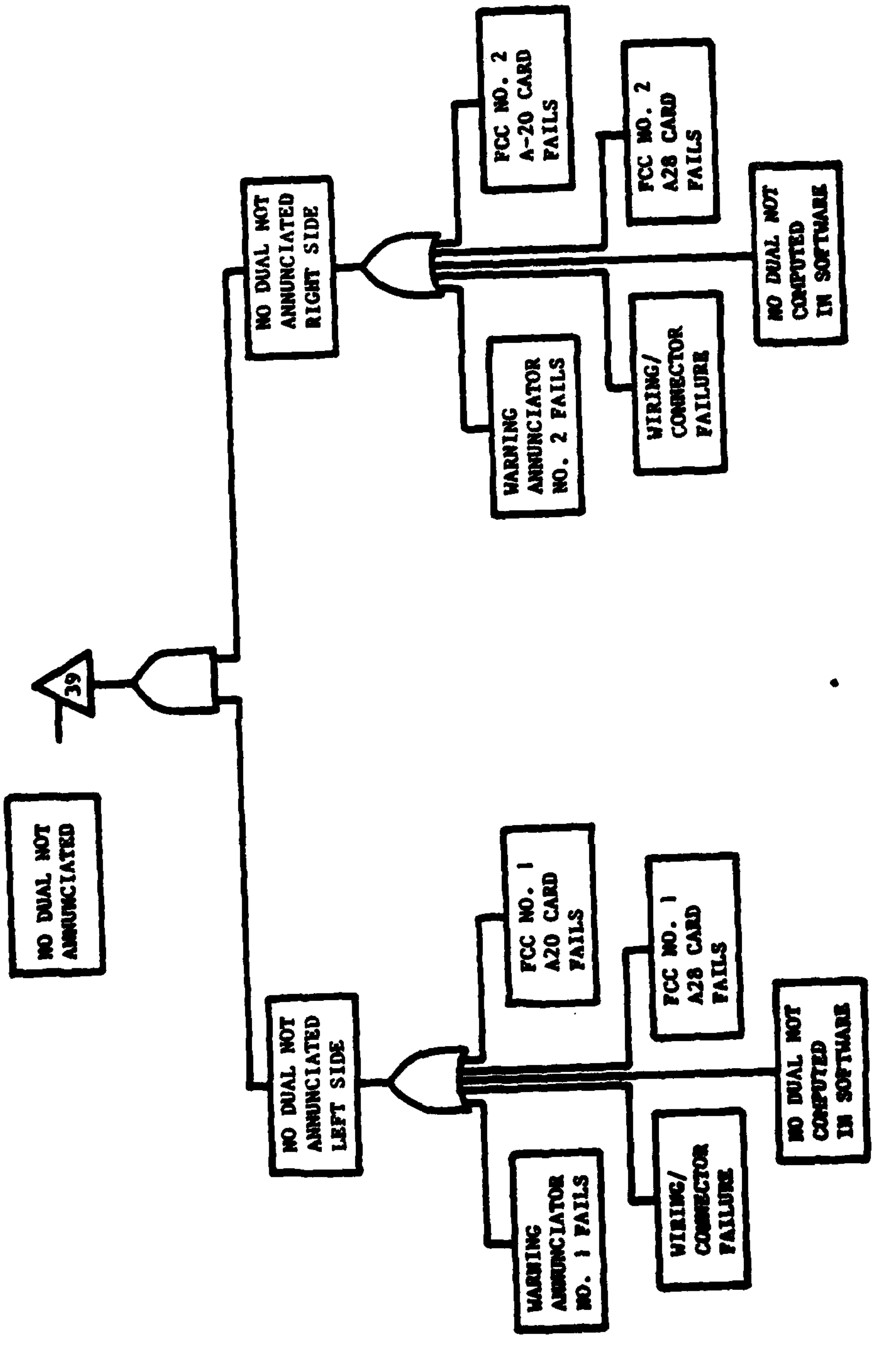


Figure 9. NO DUAL Annunciation

cause all computer channels which use that software to produce the same, but wrong, result. Multiple computer channels do not provide redundancy with respect to generic software errors as long as the same software is used in all channels, as it is in most contemporary systems, including the RDFCS. Reference 7 may be consulted for a discussion of software errors, and RTCA Document DO-178 should be consulted for a discussion of software test requirements.

Fault tree stubs similar to that shown in Figure 8 were developed for the other sensors of Figure 7. These are very much like the stub shown in Figure 8 and so are not included in the report.

The second of the Level-3 events of Figure 6 is that the crucial flight phase is initiated without fail-operational computing capability and that an additional component failure causes system failure before the phase is complete. This is shown in Figure 10 as four Level-4 events. The first of these, that channel A of FCC No. 1 fails above alert height, can be caused by either channel of the FCC failing to produce a required output, as shown by the eight events at the lowest level (Level-5) in Figure 10.

Figure 11 continues the development of the fault tree for one of the Level-5 events of Figure 10. This event, failure of the A channel of FCC No. 1 to produce a rudder command, can be caused by failure of any one of several cards within the channel. In this study, the two cards which make up the processor were considered in more depth than the others. These two, the A13 Control Card and the A14 Data Path Card, are shown in Figures 12 and 13, respectively, in terms of the modules described in Section 5.1.1.1, Volume II. Also shown in each of Figures 12 and 13 is a subevent for failure of a miscellaneous part, such as the circuit board, the edge connector, or other part which is not included in one of the modules named in the other blocks.

Theoretically, the fault tree analysis of the failure of the processor to compute the rudder command can be continued below the module level to the individual integrated circuit pins or discrete piece-parts. The desirability of doing this is questionable, however, because of the nature of the processor. The processor is not designed to perform a single specific function, such as computing rudder commands. It is designed to efficiently perform a number of simple functions, such as addition,

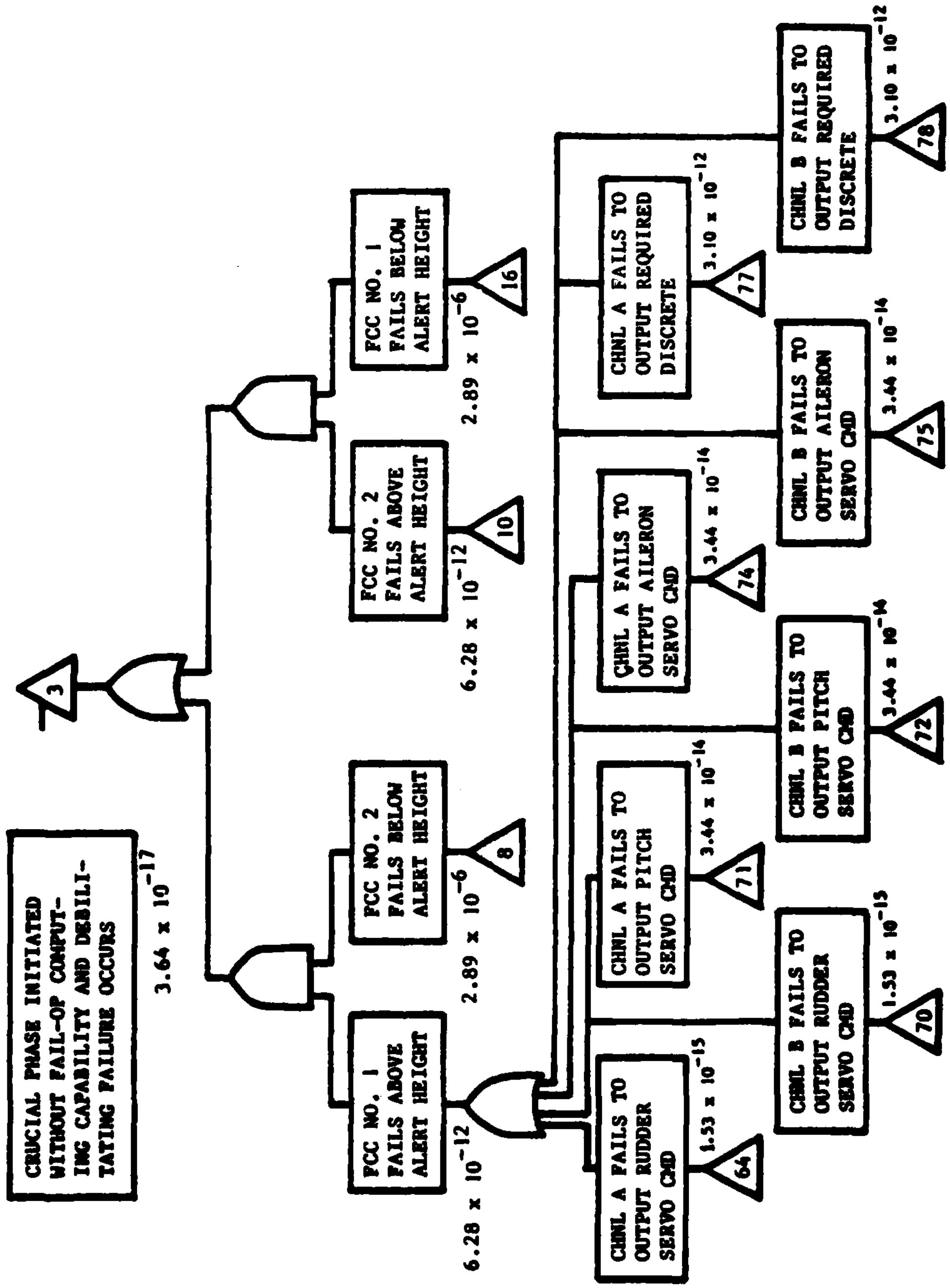


Figure 10. Computing Function

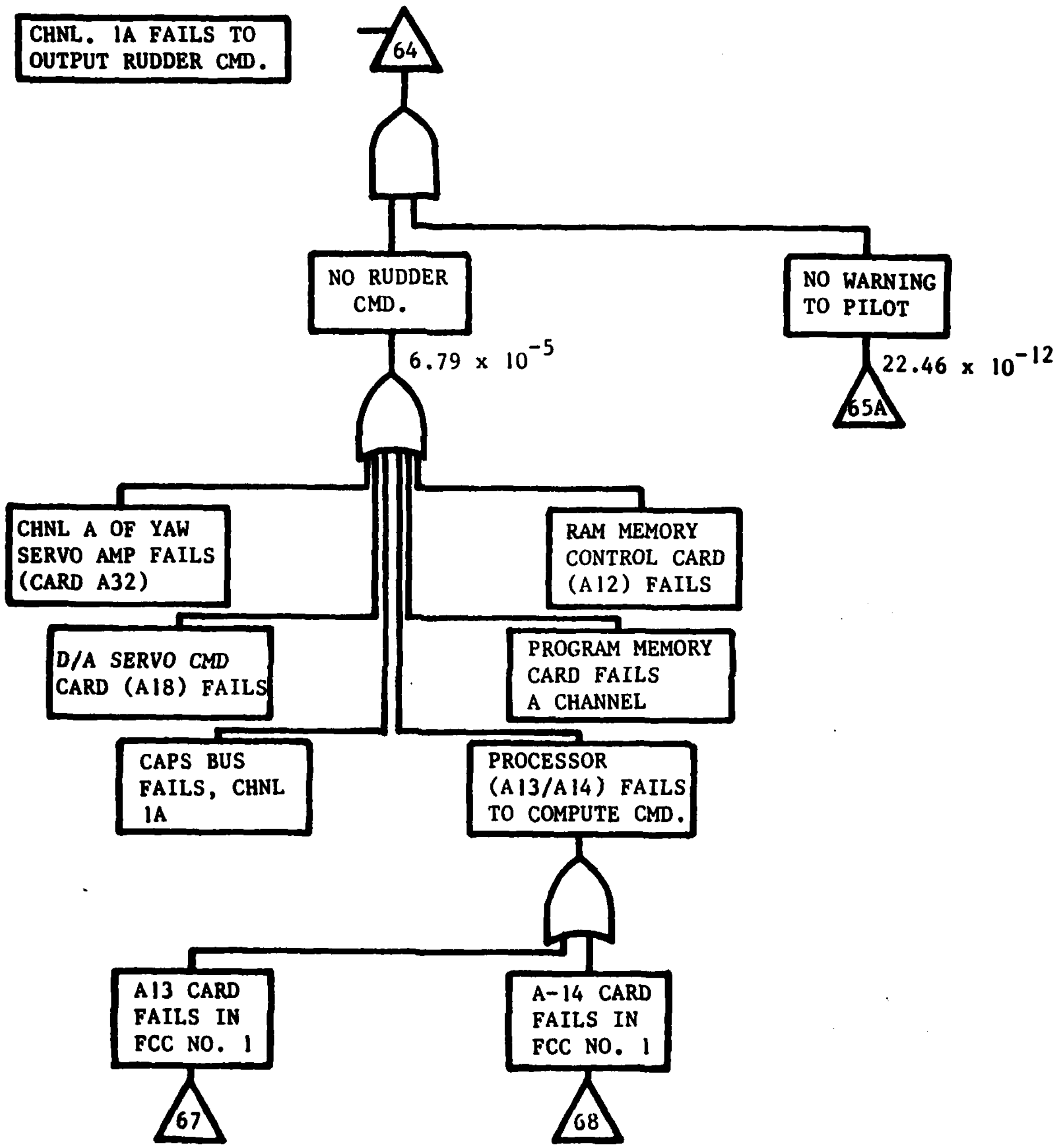


FIGURE 11. CHANNEL 1A RUDDER COMMAND

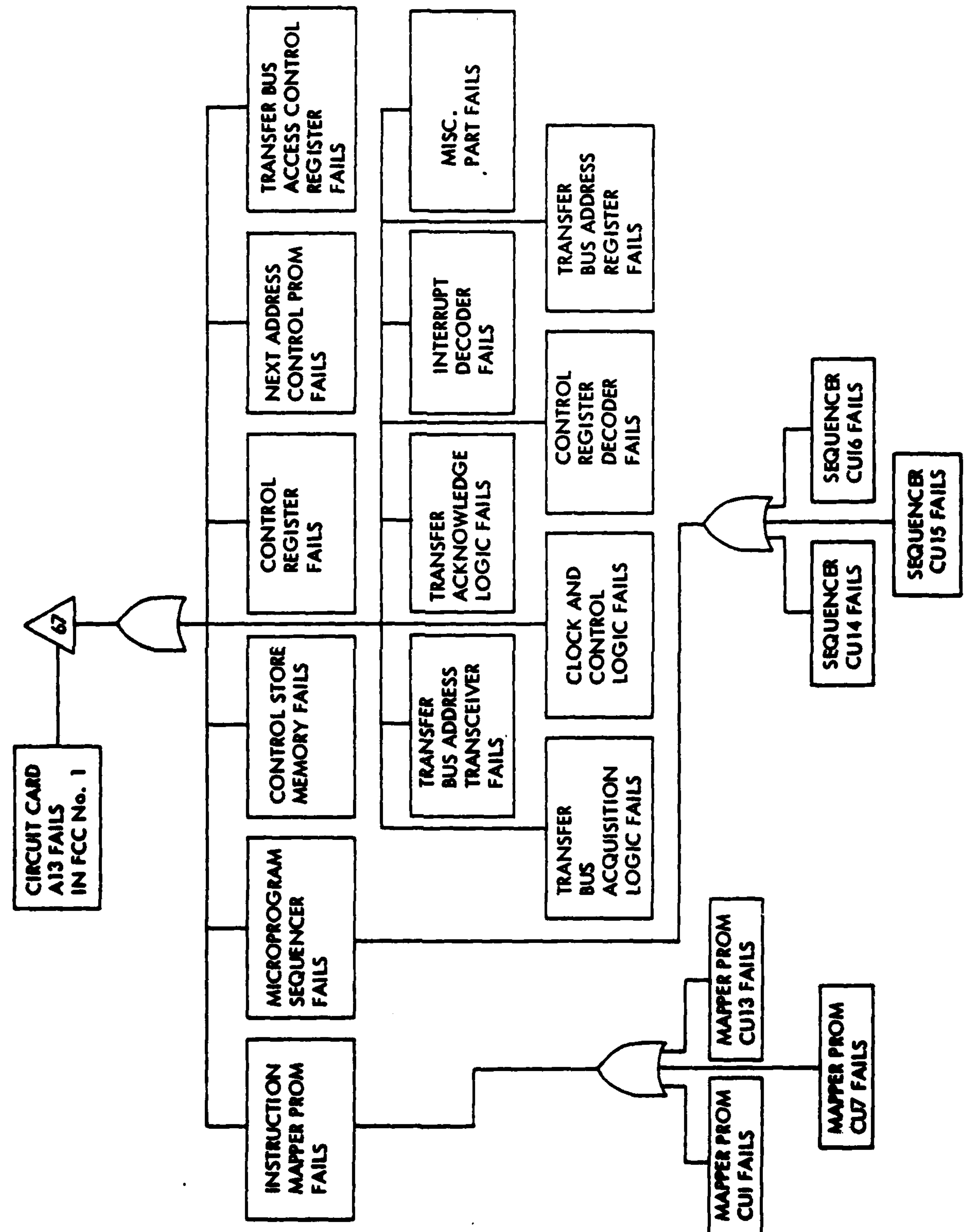


Figure 12. FCC Processor Control Card

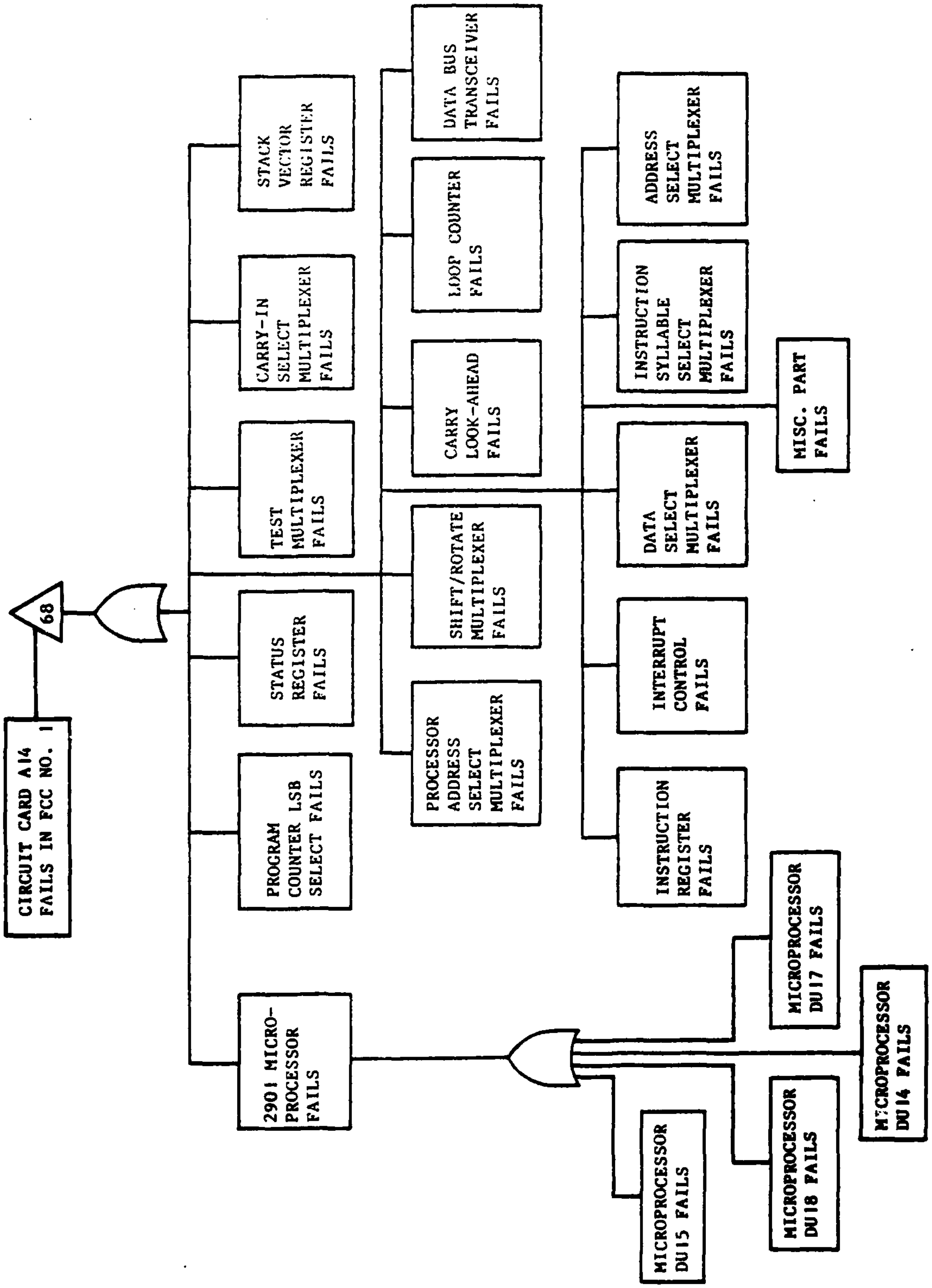


Figure 13. FCC PROCESSOR DATA PATH CARD

multiplication, and logic operations. A suitable sequence of such operations (i.e., the flight software) is used to make the processor generate the rudder command, the aileron command, and so forth. It is much easier to relate the modules and integrated circuits (IC) to the simple functions (add, multiply, etc.) than to the much more complicated functions of computing the command for a particular servo.

It is also easier, in general, to relate a specific failure mode of an integrated circuit within the processor to its effect on the processor operation than to start with the effect and then work in the other direction to the IC failure modes which would produce the effect. In other words, it is easier to do an FMEA than a fault tree analysis at this level.

Another reason for preferring FMEA to fault trees at this level is that in the course of performing the fault tree analysis, the analyst must account for all of the ways the processor can fail; that is, all of the ways in which the processor output can be wrong.

These ways are the failure modes of the processor. Each of these modes must then be traced to all possible combinations of IC pin failures which could produce the processor failure mode. Because processors have many different possible outputs, there are a high number of ways that the output could be wrong. There is no practical way of assuring that all of these possibilities have actually been covered in the fault tree. The FMEA requires that all pin-level IC failure modes be considered. These modes are much better understood, and there are less of them, so that it is much easier to be certain that they have all been covered. This is not meant to imply that a complete pin-level FMEA is easy or inexpensive; it is neither.

In light of the foregoing considerations, the fault tree analysis of the processor was not continued below the level developed in Figures 12 and 13. Instead, the FMEA approach was used as described in Section 6.

To continue with the development of other branches of the fault tree, Figure 14 develops the event of Figure 11 that the pilot is not warned that FCC No. 1 A channel is not generating a correct rudder command. This portion of the fault tree includes several software functions. In a production program, the test requirements of each of these functions should be reviewed to confirm that they satisfy the criteria of RTCA Document DO-178 (Reference 2). In this project, conducted for illustrative purposes, this review was not made.

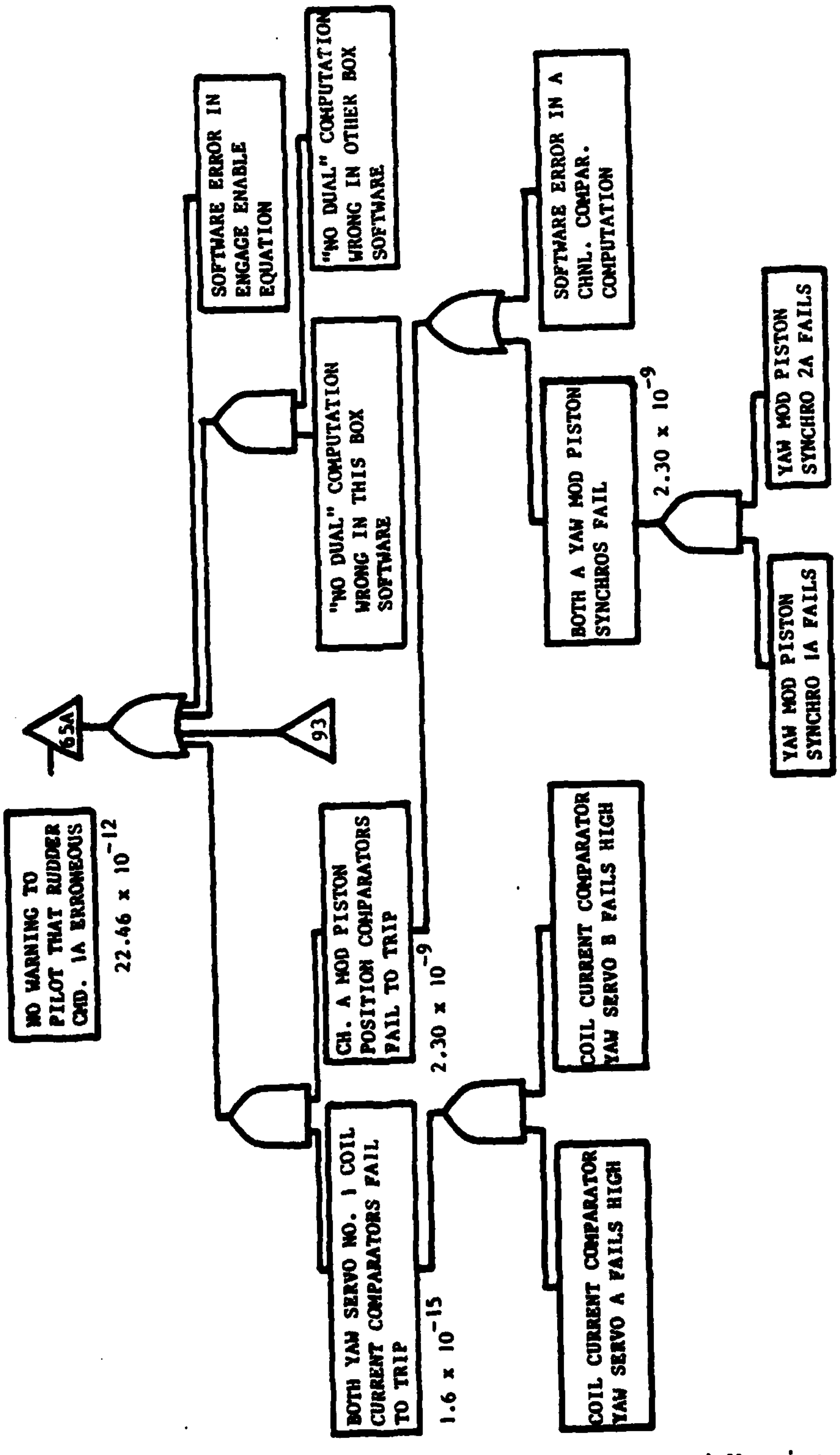


Figure 14. Yaw Autopilot Servo Command Warning

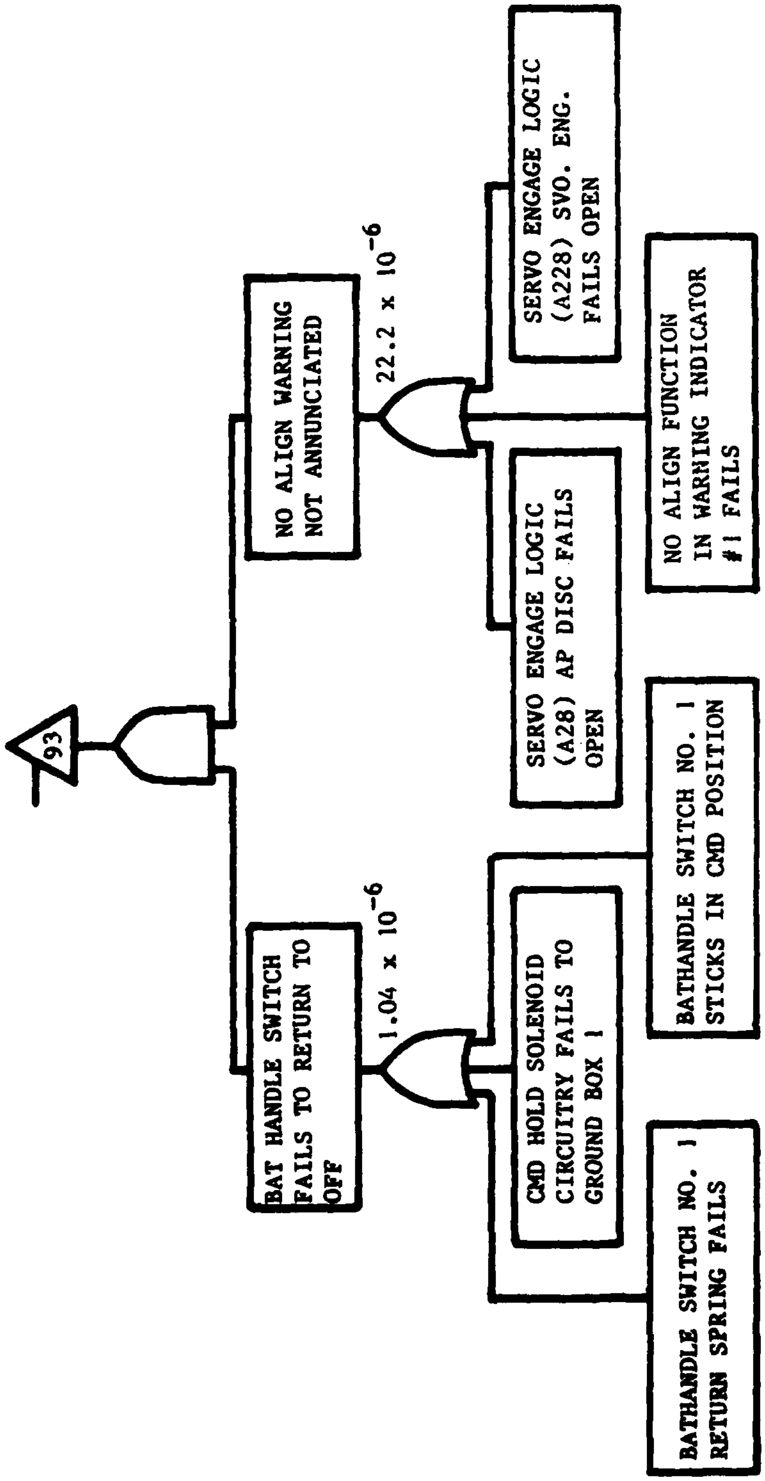


Figure 14. Yaw Autopilot Servo Command Warning

Similar tree stubs to that developed in Figures 11-14 were developed for the other required outputs from Channel A of FCC No. 1 and the other three channels (Figure 9). They are not included here because they are quite repetitive of the analysis shown.

The last of the Level-3 events of Figure 6 is that the crucial phase is initiated without fail-operational servo capability and a debilitating failure occurs. This is expanded in Figure 15 into the three aircraft control axes: roll, pitch, and yaw. Figure 16 shows the fault tree for failure of the No. 1 yaw autopilot servo, with the servo failure not annunciated to the crew.

Fault tree stubs for the other 5 servos of Figure 15 were developed to complete the analysis of the Level-3 events of Figure 6. These are quite similar to the stub shown for the rudder servo and are not included in the report. This completes the discussion of the second of the Level-2 events of Figure 6.

The third of the Level-2 events of Figure 6 is that multiple failures occur during the crucial flight phase and these occur in a combination which causes system failure. Figure 17 shows the initial development of this event to lower levels. Continuing this development produces a major branch of the fault tree quite similar but simpler to that for the second of the Level-2 events. It differs primarily in that the NO DUAL annunciation does not appear, since that particular warning is suppressed during the crucial phase. Since that major branch is so similar to that already discussed, it is not describedd further here.

QUANTITATIVE FAULT TREE ANALYSIS

System failure probability was computed from the fault tree using the hardware failure rates presented in Section 8. A failure rate of zero was used for each software function, since there is currently no acceptable way of predicting DFCS software failure rates (Reference 2, Section 2.2.1).

Considering hardware failure modes only, the probability of initiating the crucial phase with less than fail-operational capability and a second failure debilitating the system was calculated to be 2.46×10^{-14}. This is based on a flight time of 4.0 hours prior to the crucial phase and a crucial phase duration of 0.02 hours.

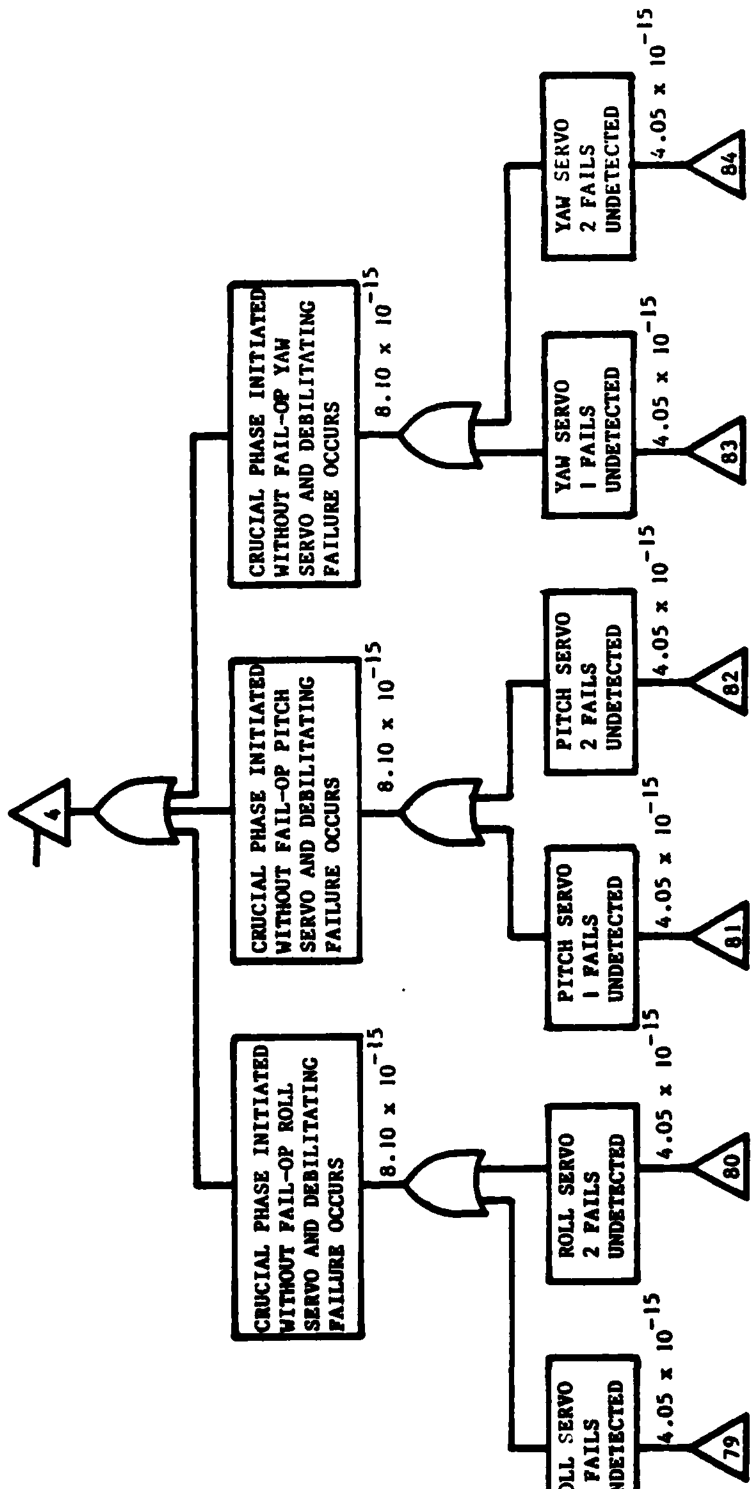


Figure 15. Servo Functions

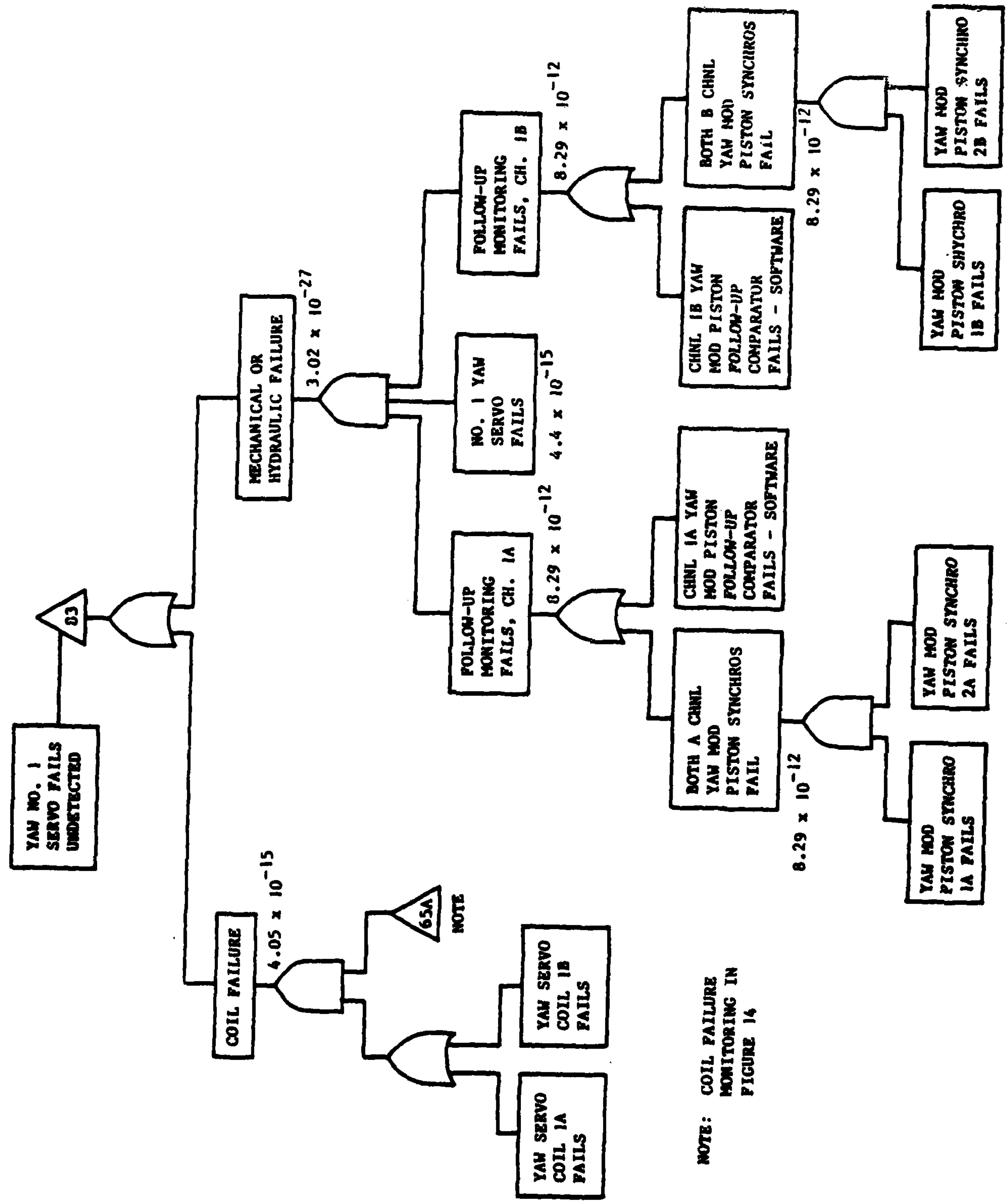


Figure 16. No. 1 Yaw Autopilot Servo

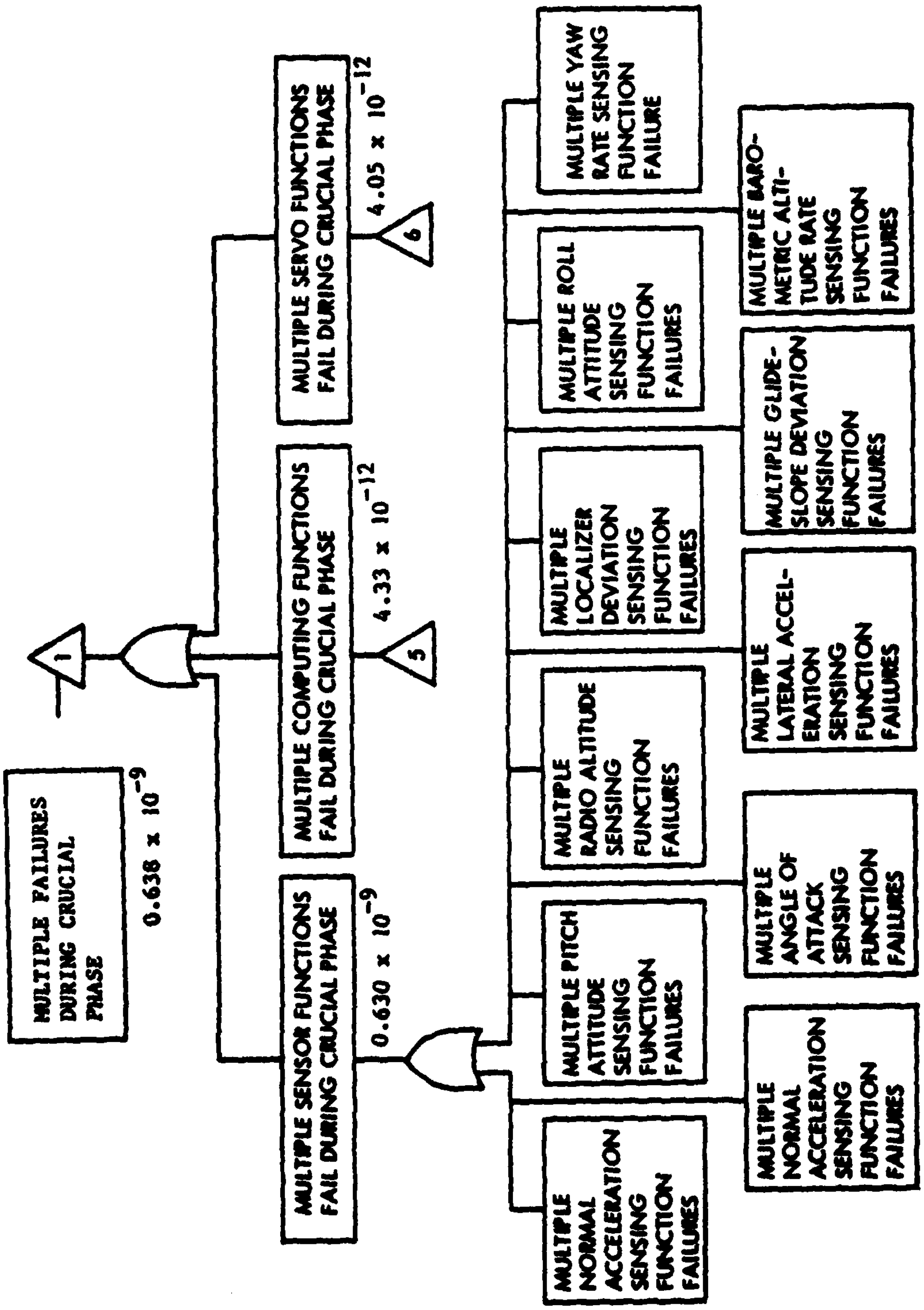


Figure 17. Multiple Failures During Crucial Phase

The probability of the system failing because of multiple failures during the crucial phase was calculated to be 0.638×10^{-9}. This is based on a crucial phase duration of 0.02 hours.

The system failure probabilities computed are actually upper bounds on the actual failure probabilities. This is because the fault trees are based on the assumption, for many items, that all failure modes of the item render the item incapable of performing any of its functions. For example, certain buffers on the A26 Data Acquisition Card are used for sensor data which is not required for automatic landing, and so at least some of the failures of these buffers would not prevent the card from correctly handling required data. However, the failure rates used in the analysis are for the entire card, including these buffers, so that the failure probability calculated for the card includes card failure modes which would not affect automatic landing.

TABLE 2. QUANTITATIVE RESULTS

Probability Of	Fault Tree Result	CARSRA Result
Unannunciated Failure in Cruise and Second Failure in Landing	2.46×10^{-14}	3.36×10^{-14}
Multiple Failures In Landing	0.64×10^{-9}	0.66×10^{-9}

6. FAILURE MODE AND EFFECT ANALYSIS

ROLE IN INTEGRATED ASSURANCE

As stated in Section 5, fault tree analysis provides assurance that most system components, such as analog sensors and servos, have no single failure mode which produces system failure. This is because such components have only a few possible failure modes, and it frequently is not necessary to distinguish in the fault tree among these modes. When it is necessary to distinguish among modes, it is usually fairly simple to identify the modes which are relevant in the branch of the tree being developed. The analysis can often be extended below the component level to the failure modes of the individual piece-parts which comprise the component. Analysis to this very detailed level is sometimes necessary to ascertain that a component has no failure modes which could remain undetected until a second failure occurs elsewhere in the system.

Fault tree analysis is cumbersome and inefficient if extended from system level to the integrated circuit pin level in the processor of a digital system, however. Basically, this is a result of two basic characteristics of digital systems:

1. Functions which are described very simply at a higher level (e.g., sensor monitoring) require a myriad of sequential operations at the integrated circuit level. These operations are required to obtain the proper data, route it to the proper registers within the arithmetic logic unit (ALU) where arithmetic and logic operations are actually performed, and route the results too the proper storage register or output port. Many different integrated circuits are involved in each of these operations.
2. Many interfaces between integrated circuits involve several pins,and it is the combination of pin states (electrically high or low) which is significant. That is, each combination of pin states represents a different data value or instruction, and the effect of a single pin being in the wrong (faulted) state depends on the state of the other (non-faulted) pins.

The net result of these characteristics of digital hardware is that there are many more integrated-circuit-level operations performed in executing the flight software than there are pin-level failure modes. In extending a fault tree analysis from failure of system-level functions to failure of integrated circuit pins, all of these detailed operations must be included and accounted for, an extremely inefficient process. Once the fault tree had been fully developed, another extremely laborious task would remain: reviewing the tree to make certain (1) that all of the failure modes of the integrated circuits had been accounted for, and that no failure mode could remain undetected until a second failure occurred, with the combined effect of both faults producing a hazardous condition; and (2) that no failure mode could by itself produce a hazardous condition.

Failure mode and effect analysis provides a means of systematically examining all of the potential failure modes of the integrated circuits to confirm that none of them could cause a hazard directly or remain latent and subsequently cause a hazard in conjunction with a second failure.

GENERAL CONSIDERATIONS

In conducting the pin-level failure mode and effect analysis of a processor, three factors greatly reduce the effort. The first factor is that propagation of most faults under all conditions does not have to be considered. A single effect can usually be found which will totally debilitate the processor. For example, a faulted processor output pin will result in the processor trying to read about half of the data and machine level instructions from the wrong memory addresses. This will result in the coil current comparators tripping, sensor comparisons failing, and in the case of the RDFCS, the iteration monitor will fail. In a system using check-sums to monitor program memory integrity, these tests will fail.

The second factor which reduces the effort is that many pairs of faults will have the same effect. There are numerous instances of an output pin on one IC being connected only to one other pin. If either pin fails open, the effect will be the same. Similarly, a ground fault in either pin will produce the same effect.

The third factor which reduces the effort is that there are many instances in which three pins are connected so that one output pin drives two input pins on different circuits. An open fault at each of the input pins can be evaluated first. An open fault at the output pin is then equivalent to both input pins failing open simultaneously, and in most cases the effect is the "sum" of the effects of the input pins failing open; that is, both effects occur. If both input pins are on the same chip, the effect of both being open is more likely to differ from the sum of the individual effects. See Figure 18.

The effect of any of the three pins failing shorted to ground is the same in either of the two cases of Figure 18.

Another frequently encountered condition involving three pins is two outputs connected to a single input (Figure 19). In such a case, chips A and B will have three-state outputs, and one or both outputs should be in the high-impedence state at all times. An open fault on the output pin of chip A will then only affect chip C when A has its output enabled. Similarly, an open fault on the output pin of chip B will only affect chip C when B has its output enabled. An open fault on the chip C input pin will usualy produce the sum of the effects of open faults on the two output pins. A ground fault on any of the three pins will have the same effect.

Still referring to Figure 19, if a fault should occur which results in both enable pins being in the enable state, there is a possibility of damage to the A or B chip. If one output is high and the other low, there could be a low impedence path to ground, through the output pins, which could burn out the A or B chip. This depends on the technology used in the individual chips. Frequently, the effect of the original ground fault can be judged to be a total processor failure whether or not the secondary damage occurs.

APPLICATION OF RDFCS

In this study, three modules of the processor (Figure 20) were considered at pin level (Ref. Vol. II, Section 5.1.1.1):

- The instruction mapper prom, which consists of three prom chips in parallel

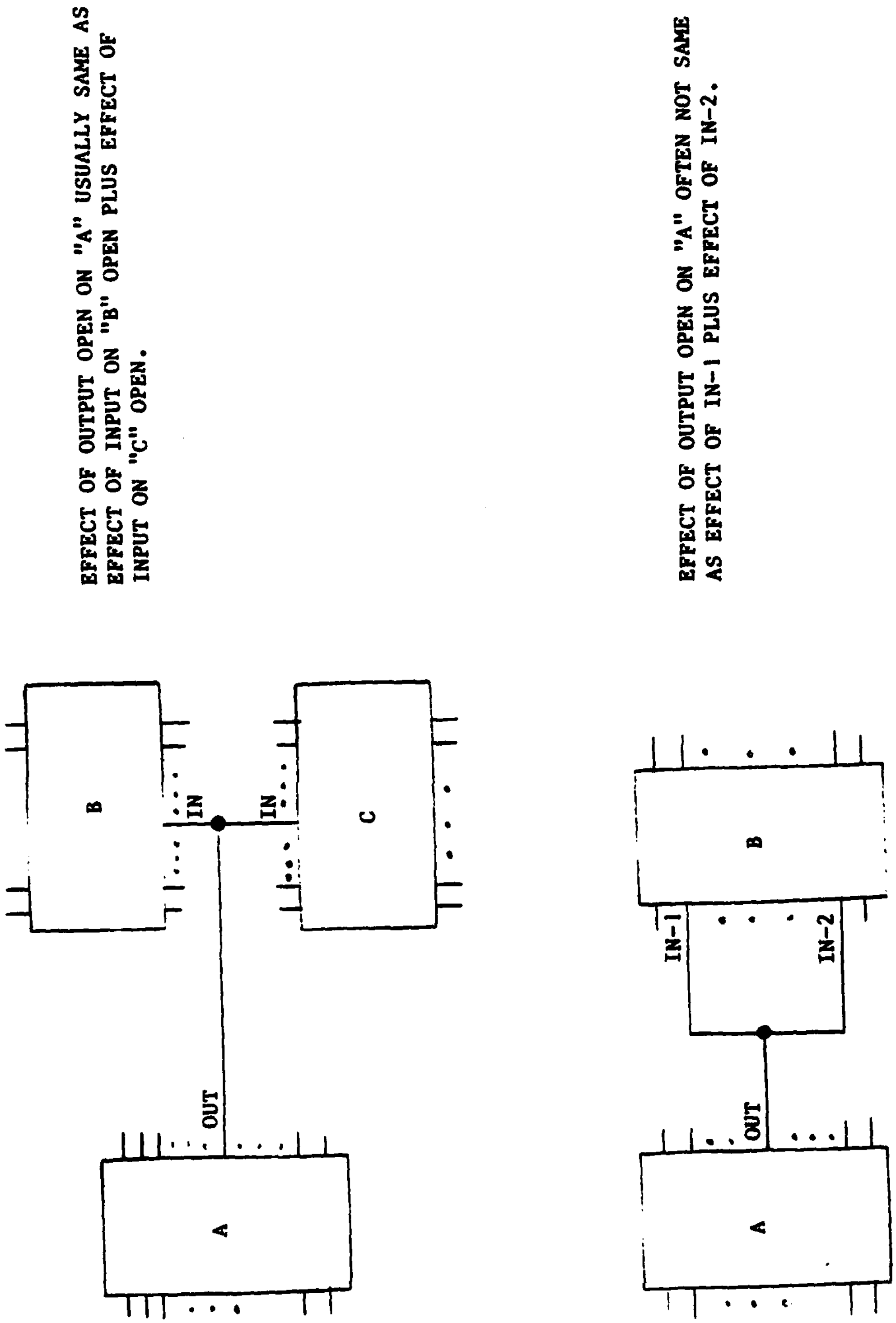


Figure 18. One Output, Two Input Conditions

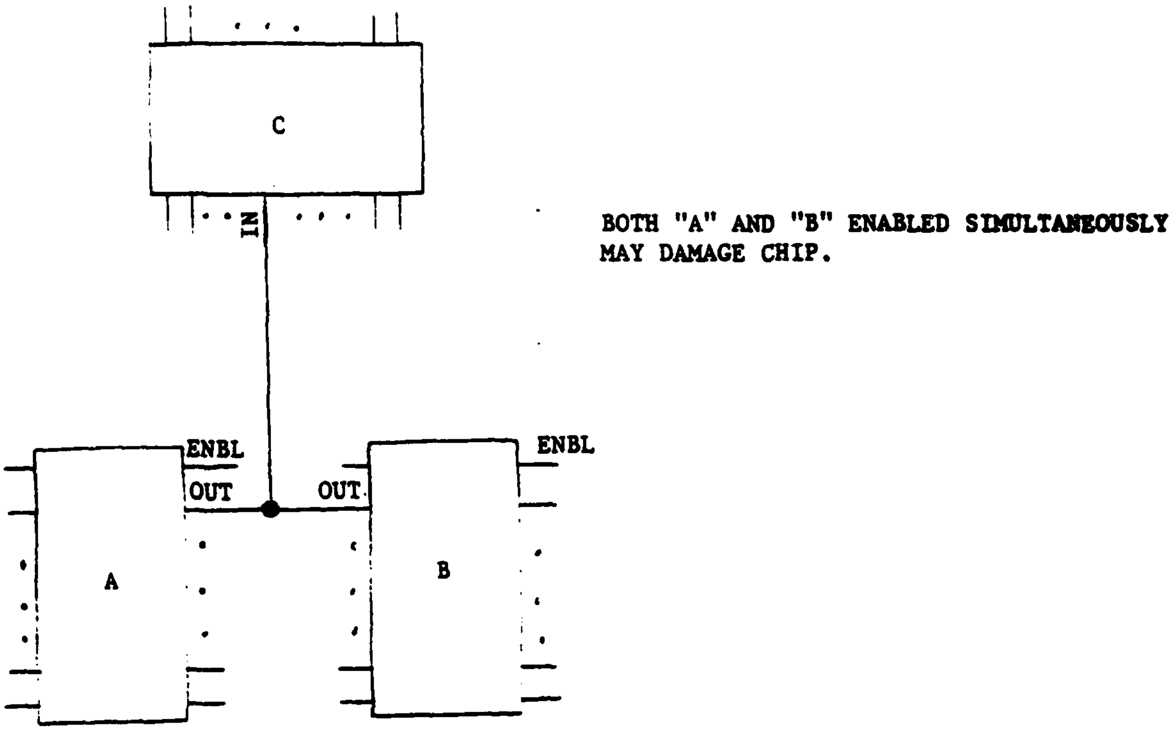


Figure 19. Two Output, One Input Condition

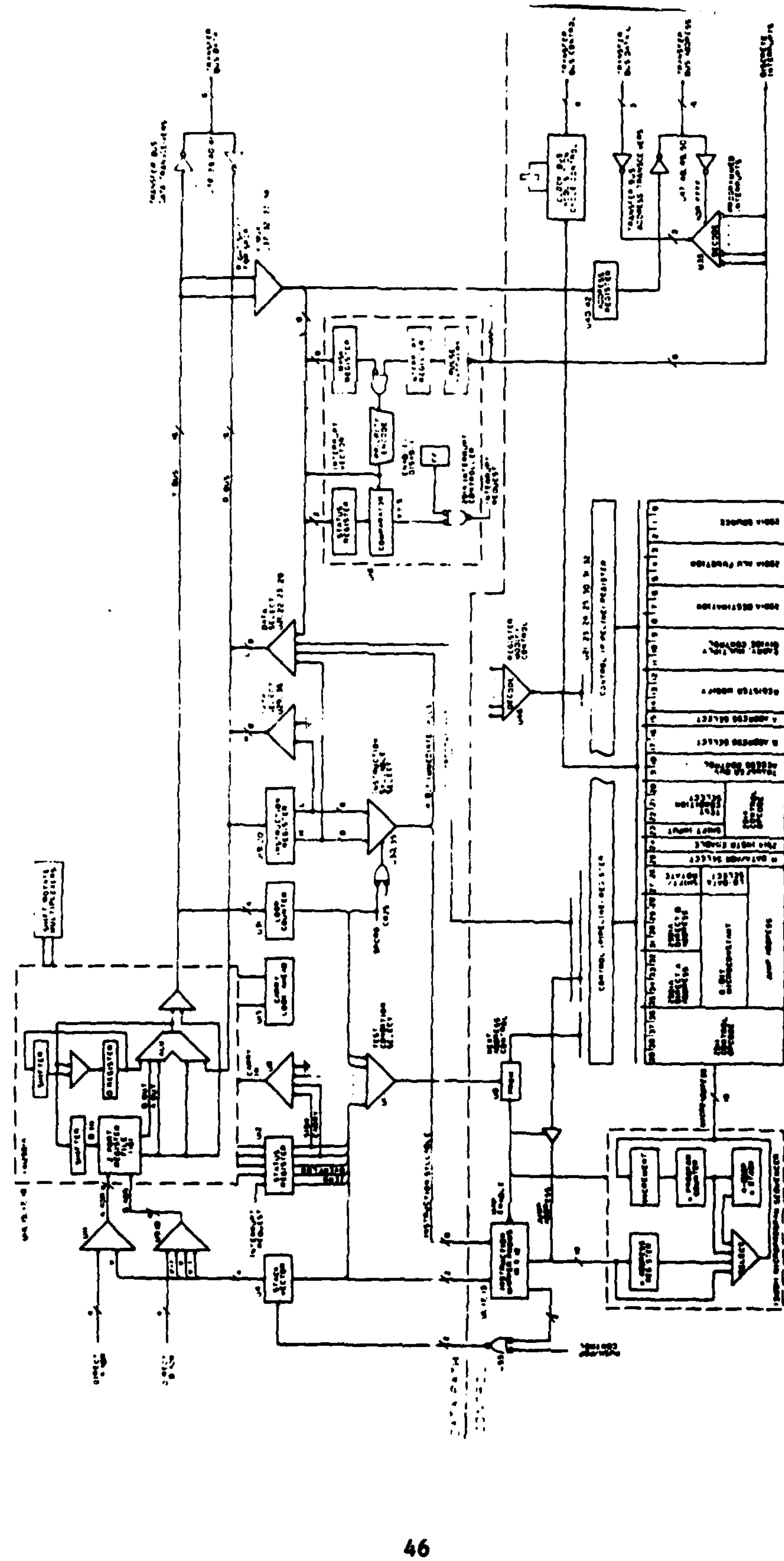


FIGURE 20. PROCESSOR BLOCK DIAGRAM

- The microprogram sequencer, which consists of three 2911 sequencer chips in parallel

- The microprocessor module, which consists of 4 chips in parallel. Each of these chips is a 2901A.

The instruction mapper prom chips are read-only memory chips. The inputs to the chip are machine-level operation codes and the depth of the stack maintained in the 2901 microprocessors. These are connected to the address pins of the mapper. The data stored in the prom is the control store prom address of the first microcode instruction required to execute the machine level instruction with the processor stack at a particular depth. The mapper output pins are only active at the beginning of a microcode sequence, at which time a chip enable signal is sent to the mapper from the next address control prom.

The microcode address from the mapper prom is routed to the microprogram sequencer module. This module generates a sequence of microcode addresses, beginning with the starting address from the mapper prom. Some microcode routines involve jumps to a new address rather than sequential progression only. In such cases, the microprogram sequencer receives the jump address from the control store proms and resumes sequential generation of addresses.

The microprocessor module is composed of four 2901A microprocessor chips. Each chip has a word size of 4 bits, so that the four chips in parallel are used to provide the processor 16-bit word size. This requires that carry signals be passed between 2901A's during arithmetic operations. Other interconnections between 2901A's are used for data shift operations.

The 2901A's are controlled primarily by control signals from the control store proms in conjunction with the outputs from various registers. Section 5.1.1.1 of Volume II should be consulted for further information on the functions of these registers and other processor modules.

The failure mode and effect analysis, summarized in Table 3, (in Appendix A) considered three types of pin-level faults: open, grounded, and shorted to supply voltage. In most cases, the effect of a fault can be assessed by using the chip logic diagrams, a description of chip/module functions and the schematic diagrams (Volume II, Sections 5.1.1.1. - 5.1.1.5). The schematic diagrams are reproduced in Appendix C.

The effect of certain pin faults cannot be determined by analysis using just the information mentioned above. In particular, the contents of specific prom addresses is needed in some cases. In other cases the machine-level code is needed along with the microcode sequences and addresses. Alternatively, the faults can be inserted and the effect observed. This approach was taken in this study and the results are presented in Section 7. For example, it was known that failure of one of the processor pins used in data shifts (R0, R3, Q0, Q3 stuck high or low), there would be an immediate disconnect if certain of the integer words made up of packed Boolean variables were shifted. It was determinable from the available information that such shifts might occur, but it was not determinable that they definitely would occur. Volume II, Tables 5.1.4.3.3.3 and 5.1.4.3.3.4 show examples of such packed words. Similarly, if certain fixed-point numbers were shifted during computation, the commands to the servos would be in error and the coil current comparators would trip. While both left and right-shifts are normally used in multiplication algorithms, it was not determinable that a stuck shift bit would definitely cause such a trip. When the faults were actually inserted, the processor stopped immediately. ("Immediately," as viewed by the human observers.) In this way, fault insertion confirmed the overall effect, massive processor failure and disengagement of the servos, but the exact mechanism by which it occurred was not determined.

7. FAULT INSERTION

ROLE IN INTEGRATED APPROACH

Fault insertion is used in the integrated assurance approach for three purposes as shown in Table 1. These are:

1. Faults are inserted, on a sampling basis, to confirm the fault effects reflected in the fault tree analysis and fault effects determined during failure mode and effect analysis. This includes faults of components (sensors and servos in this study) and faults of integrated circuits (pin-level faults in the digital processor).

2. Faults are inserted, also on a sampling basis, to confirm fault detection and annunciation functions implemented in the system. Many of these are also inserted to confirm effects, so that they are inserted for two specific purposes.

3. Faults are inserted to determine the effect when the analysis is intractable or when there is some uncertainty in the analysis result.

APPLICATION TO RDFCS

The RDFCS simulator at NASA-Ames was used to insert the faults shown in Table 4 (in Appendix B). The faults were of two general types: component level faults and integrated circuit pin faults. The component level faults were inserted using the FCC breakout panels (Figure 21), the Servo Simulator Panel (Figure 22), and the MDICU. Single-sensor faults are those numbered 1 through 19 in Table 4.

Faults representing a dead sensor or a broken wire from the sensor to the FCC were inserted by pulling the appropriate jumper plug at the breakout panel. Faults representing missing sensor validity discretes were also inserted in this way, although they can also be inserted via the Discrete

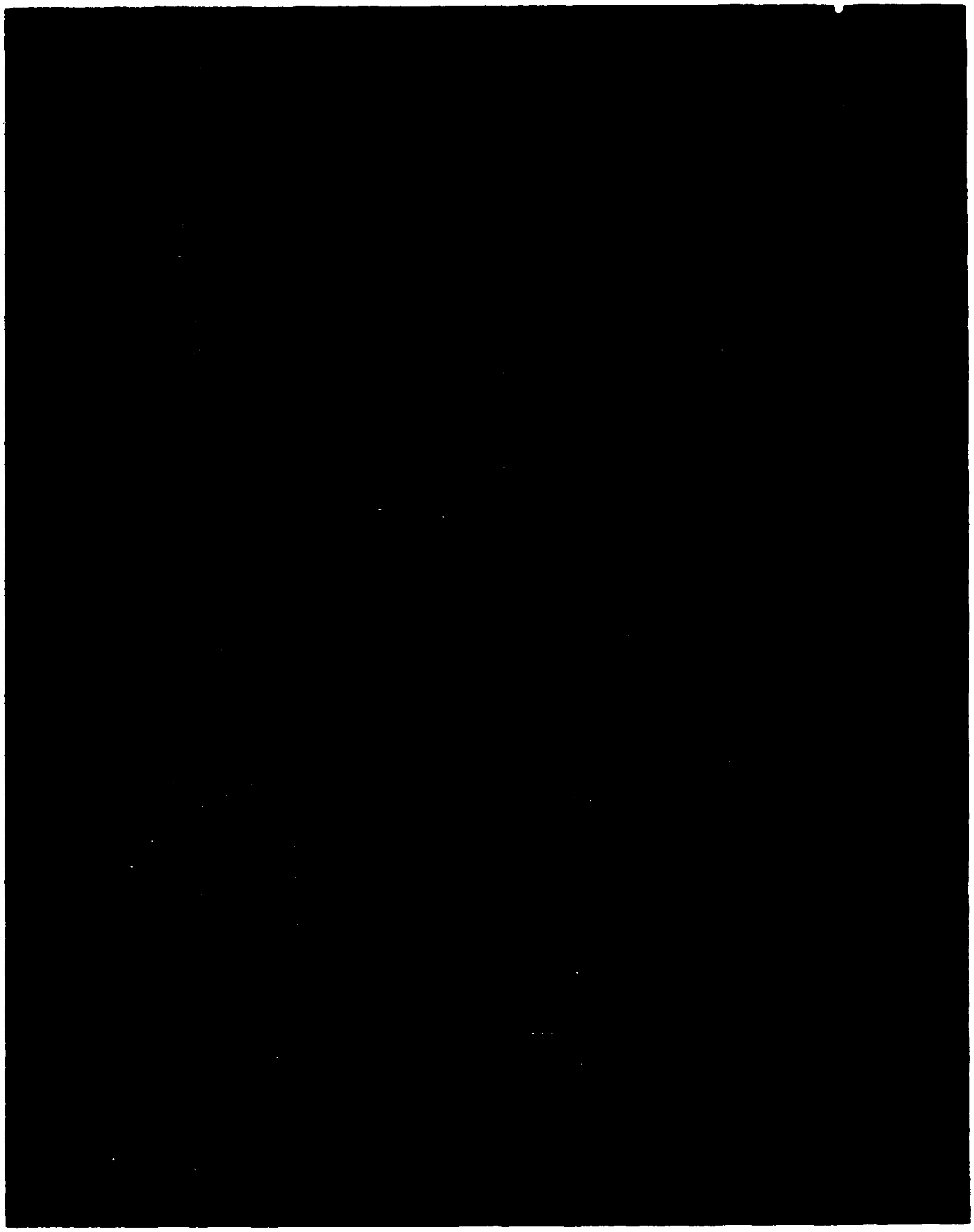


Figure 21. CAPS Test Adapter and Computer Breakout Panel

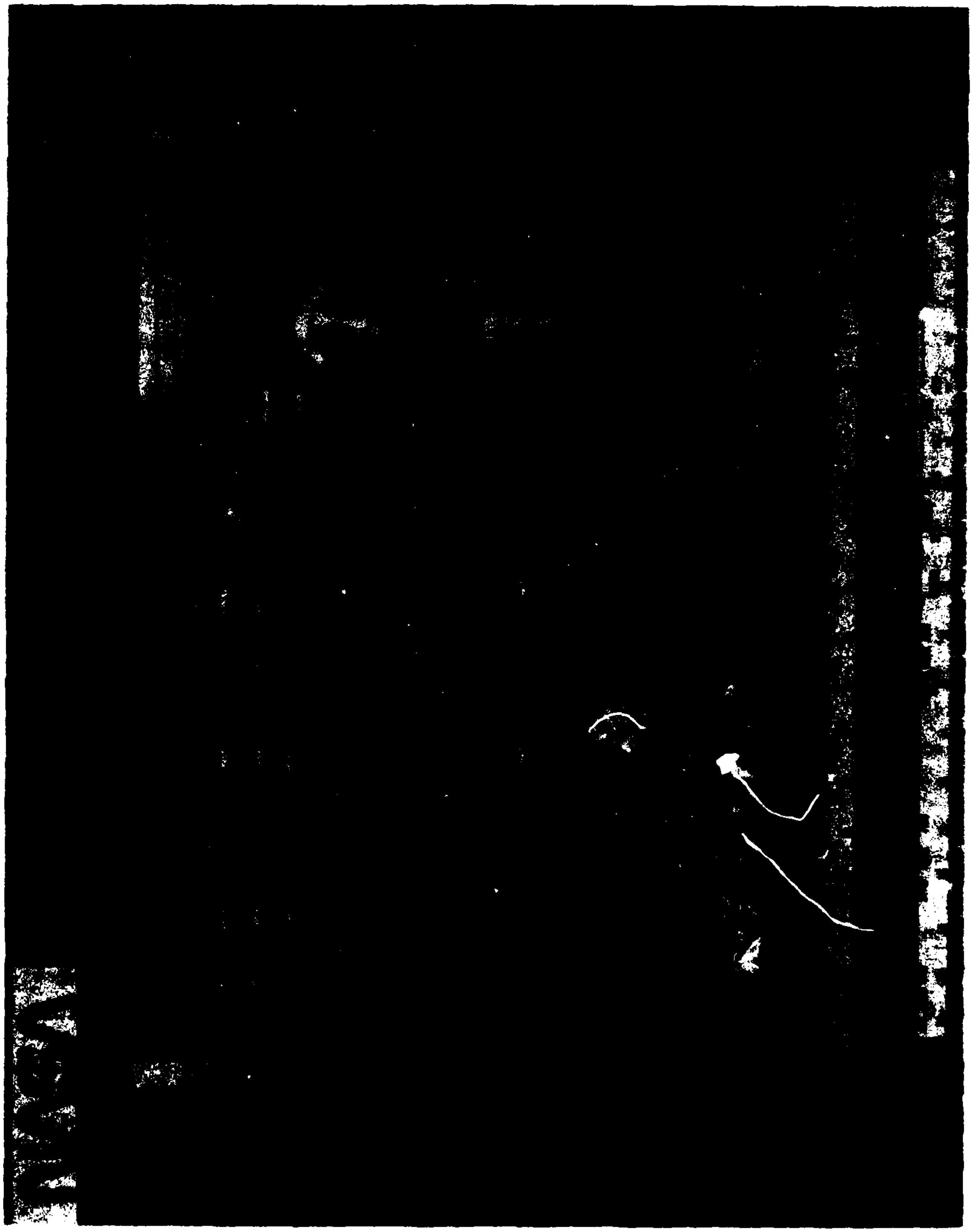


Figure 22. Servo Simulator Panel

Switch Panel (Figure 23). Sensor hardovers and ramps were inserted using the MDICU. Servo faults were inserted using the Servo Simulator Panel.

For monitoring the processor detection of sensor faults, the CAPS test Adapters (CTA) were used. One of the CTA address windows was set to the adddress of the Executive Failure (Status) Word (EFW) in each computer channel. The EFW is a 16-it word with each bit representing a discrete piece of information and there is one EFW for each sensor type in each computer channel. The 4 low-order bits (0-3) represent respectively failure of the My A (EFMA), My B (EFMB), Other A (EFOA), and Other B (EFMB) sensor signals. The other 12 bits have functions as described in Volume II, Table 5.1.2.4.2, which are not of concern here. The data window of the CTA shows the status of the EFW as four hexadecimal characters, with the right-most character representing the bits of interest, 0-3.

The effect of a sensor signal being detected bad by the software sensor monitor is that certain bits are changed from 0 to 1. With no failures detected, EFMA, EFMB, EFOA, and EFOB are all 0, which is represented in hexadecimal notation as 0. (0000 binary = 0 hexadecimal.) When the number 1 sensor of a triple sensor complement is detected to have failed, bit 0 (EFMA) is set to 1 in both channels of FCC No. 1. Bit 1 is also set to 1 so that the comparison monitoring will work properly on the two remaining sensors. The EFW low order bits will then be 0011, which is 3 in hexadecimal. The net effect, then, of the number 1 sensor of a triple sensor set failing is that the value displayed in the CTA window changes from 0000 to 0003. The left-most three hexadecimal digits each remains at 0 since each of the corresponding binary bits (4-15) of the EFW remains at 0.

Fault cases 1 through 8 were used to show that the software sensor monitor subroutine is implemented correctly in the RDFCS by subjecting it to a number of different faults in the same sensor type. These cases were also used to show that the results of the sensor monitoring are accounted for in the implementation of the NO DUAL equation, which is also in software. Cases 9 through 16 were then used to show that the voter is involved for various sensor types. Rigorous validation of the system by testing would require that faults be inserted for all sensor types used in automatic landing. In this study, performed for illustrative purposes, the full complement of sensor types was not faulted.

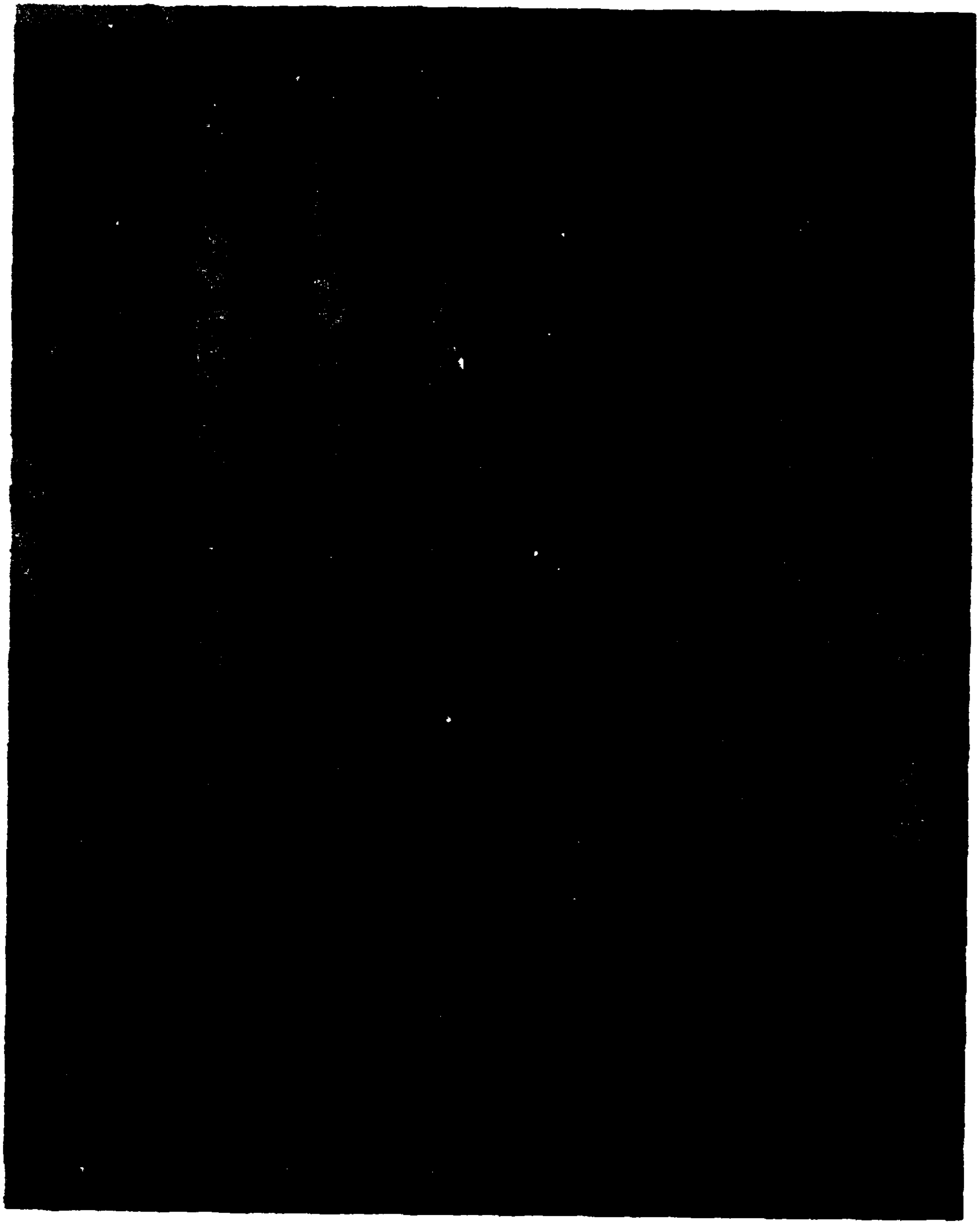


Figure 23. Discrete Switch Panel

In case 2E, NO DUAL did not annunciate even though the fault was inserted with the airplane inbound to the ILS beam intercept point. It is believed to be the result of the inbound leg being flown at an unrealistically low altitude, so that the airplane did not track the glideslope beam for 25 seconds before passing through 150 ft altitude. A review of the NO DUAL annunciation logic (Volume II, Section 5.1.2.3.1.3) shows that this is the most likely cause, since AP.ONEFAIL was set to true. Low approaches (1500 ft) were being simulated in the interest of time. Approach altitude was subsequently raised to 2000 ft.

Faults 17 through 19 were used to confirm the servo monitoring and the tie-in of the servo monitor outputs to the NO DUAL and disconnect logic. The servo monitors, in particular the coil current comparators, are quite important in ensuring that the airplane does not enter the crucial phase with a faulty computer or servo.

Fault cases 43 through 45 were used to confirm that the FCC's will both disengage upon loss of the second sensor, with the AP.DISC warning displayed, in accordance with the system description, Volume II, Section 4.3.6.1.

At the integrated circuit pin level, a number of open and ground faults were inserted to confirm the FMEA results of Section 6. For this activity, one of the FCC's was removed from the pallet and the card containing the chip to be faulted was extended for access as shown in Figure 24. Figure 25 shows the processor Data Path card.

Open pin faults, Cases 20 through 23, were inserted by using multiple sockets between the chip and the circuit card, with a jumper wire replacing the normal pin-to-socket connection. Each fault was inserted by physically pulling the jumper to open the connection. This is a slow procedure, since the chip must be removed and the jumper wire rigged on the desired pin. The chip and sockets must then be installed and the processors brought back up. This means of inserting open pin faults is only marginally satisfactory. It would be much easier to do if a stack of 5 or 6 sockets could be used between the chip and the circuit card. However, the processor will not come up with more than three sockets stacked. The longer electrical paths resulting from the use of the extender cad apparently come close to exhausting the available tolerance in the timing of the individual micro-

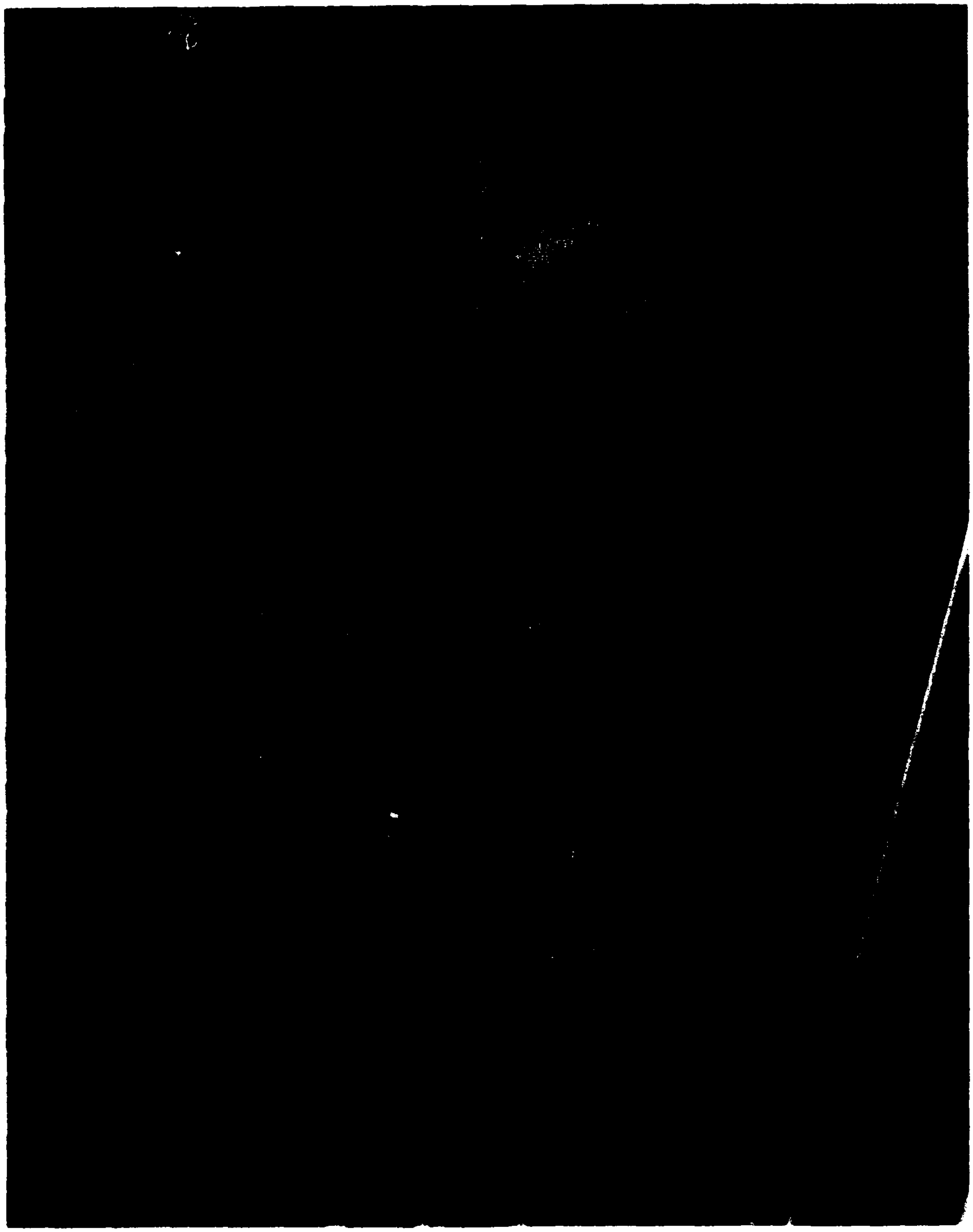


Figure 24. FCC With Processor Card Extended

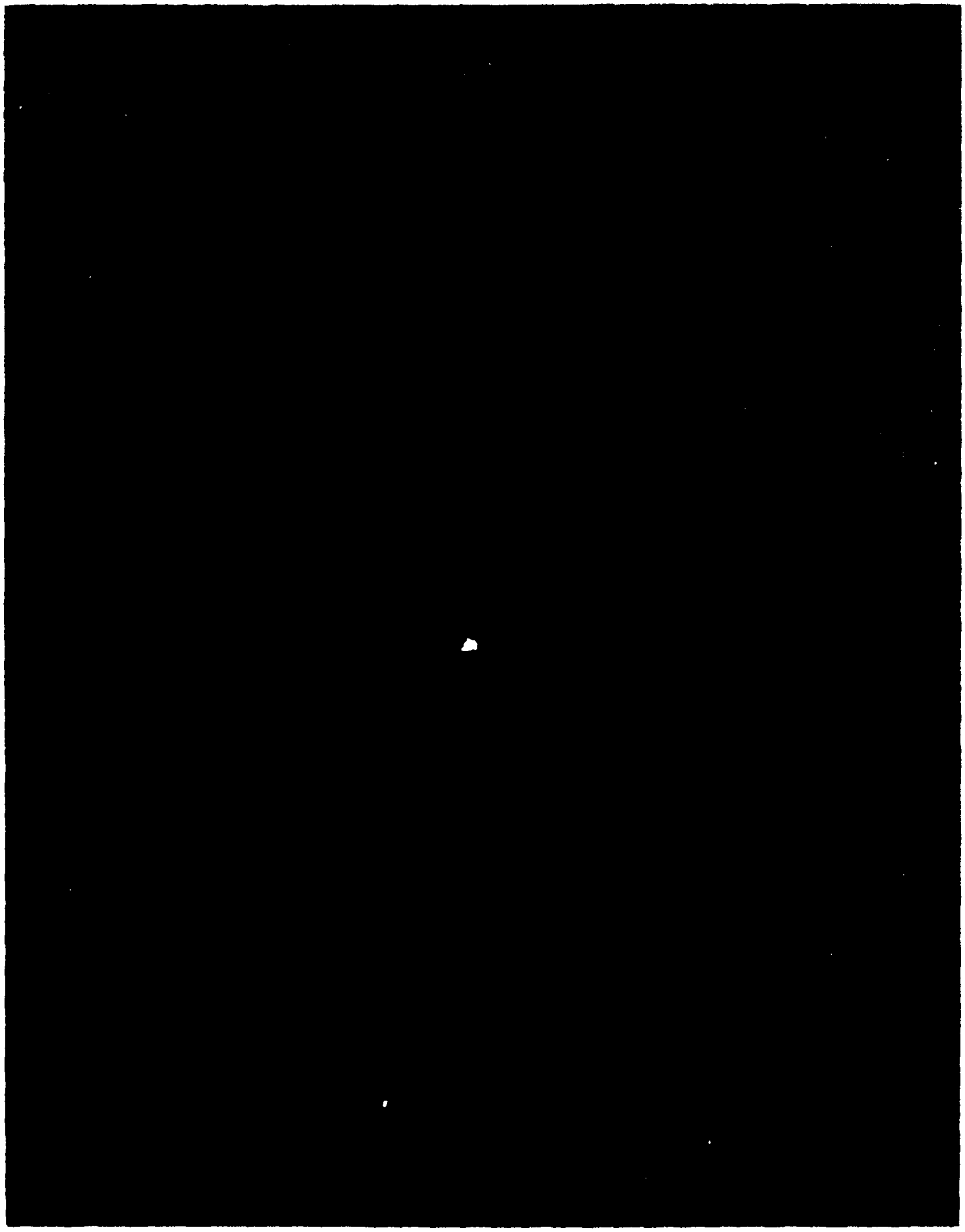


Figure 25. FCC Processor Data Path Card

steps, and the extra path length and capacitance caused by more than three sockets disables the processor.

Grounded pin faults are much easier to insert, since the chip does not have to be removed to set up each case. The processor does have to be brought back up each time, but this is a fairly rapid step. Before each fault was inserted, the data sheets from the chip manufacturer were reviewed, along with the card schematics, to determine that the fault would not damage any chips. No chips were damaged by the ground faults. The ground pin faults are cases 24 through 42 in Table 3.

The chip pin faults all disabled the processor, with the exception of open pin fault 21. This fault involves a pin of a quad 2-input NOR gate. The fault had no effect on the processor operation.

FAULT INSERTION RESULTS

The faults inserted in the RDFCS simulator achieved the desired results in the assurance assessment of this study , and more importantly confirmed that fault insertion is capable of providing the results required of it in the integrated assurance approach. Specifically, the faults inserted confirmed (1) that the NO DUAL warning appears when it should, (2) that all sensor types faulted and required for automatic landing are monitored, (3) that the servo monitoring functions correctly, (4) that the effect of pin-level faults in the processor is in agreement with the failure mode and effect analysis, and (5) that fault insertion is a reasonable way of resolving uncertainty of the effect of open and grounded pin faults in digital hardware. While these results were obtained on a particular system, the approach is judged to be viable for validating other digital systems.

8. FAILURE RATE DEVELOPMENT

The failure rates for servos, sensors, and indicators were taken from the data base maintained by the Lockheed-Georgia Company Reliability Engineering Department. They are composite values for representative components of comparable complexity and construction.

The failure rates for the integrated circuits of the Data Path and Control Cards were estimated using the formulas and tables of Military Handbook 217C (Ref. 8). The formulas provide a means of accounting for a significant number of factors:

1. Device technology
2. Device complexity
3. Junction temperature
4. Package technology
5. Applicaiton environment (voltage)
6. Usage environment
7. Quality level

For example, the equation for the failure rate of a monolithic bipolar device is:

$$f = K_0 [C_1 K_T K_V + (C_2 + C_3) K_E] K_L$$

where:

f is the device failure rate

K_0 is the quality factor

K_T is the temperature adjustment factor for junctions

K_V is the voltage derating stress factor

K_E is the applicaiton environment factor

C_1 and C_2 are complexity factors based on transistor count

C_3 is a complexity factor based on package technology and number of pins

K_L is a learning factor.

The quality factor, K_Q, has a value of 1 for devices procured in full accordance with MIL-M-38510 (Ref. 9), Class B requirements. This value was used for all circuits in this project. It should be noted that the quality factor is a direct multiplier, so that the predicted rate is proportional to it. More or less stringent quality factors can therefore greatly influence the prediction for any individual circuit, circuit board, or an entire component.

Junction temperatures are used in determing the adjustment factors K_T. The junction temperature is ambient temperature plus the differential resulting from power dissipation through the case. An ambient of 60°C was used, with the power dissipation taken from the circuit specification.

The voltage derating stress factor is 1 for the bipolar circuits used in the CAPS processor. The application environment factor is 3.5 for the airborne, inhabited, transport environment of the aircraft underdeck avionics rack. Failure rates for the circuit cards of the FCC's were obtained by summing the failure rates for the card and its components. Table 5 summarizes the failure rate prediction for the A13 control card. Failure rates for the other cards are shown in Table 6.

Table 7 presents failure rates for the system components other than the FCC's.

In using these rates in the fault tree and CARSRA analyses, an adjustment was frequently required to include only a portion of the rate, since only certain failure modes are of interest. For example, each dual current comparator has a predicted failure rate of 0.03. Each half of the comparator is given a rate of .01 for the failure mode of failing to trip when the threshold difference is exceeded. This is a very conservative rate for this mode.

TABLE 5. FCC CONTROL CARD FAILURE RATE

ITEM	FAILURE RATE*
Integrated circuits	1.788
Resistors	.0018
Capacitors	.224
Oscillator	.25
Coil	.0007
Circuit Board	.023
Edge Connector	.16
Control Card	2.45

*All failure rates in failures per million hours.

TABLE 6. PREDICTED FCC CARD FAILURE RATES

CARD NO.	FAILURE RATE*
A1 Power Supply Monitor	0.555
A2-A5 Prom Card	.809 each
A6 Power Supply Monitor	.55
A7 - A10 Prom Card	.809 each
A11 Terminator/Test Access	.555
A12 RAM Memory Control	1.18
A13 CAPS Control	2.45
A14 CAPS Data Path	1.98
A16 Cross-channel Receiver	.70
A17 DITS Transmitter	1.75
A18 D/A Servo Command	1.75
A19 Terminator/Time Synch	1.40
A20 Discrete Output	2.79
A21 Data Transmitter/Receiver	.70
A22 Serial Digital Input No. 1	1.65
A23 Serial Digital Input No. 2	1.80
A24 Autoland Sensor Input	1.80
A25 Cruise Sensor Input	1.12
A26 Data Acquisition	1.20
A27 Discrete Input	1.30
A38 Servo Engage Logic	2.61
A29 Cross Channel XMTR	1.20
A30 - A32 Servo Amplifier	3.00
A33 Speed Servo Amp	1.70
A300 Speed Command XMTR	1.70
A400 Power Supply	21.0
A500 Power Supply	21.0

*All failure rates in failures per million hours.

TABLE 7. FAILURE RATES FOR MAJOR RDFCS COMPONENTS

COMPONENT	UNIT FAILURE RATE*
Pitch Angle Gyro	303
Roll Angle Gyro	303
Yaw Rate Gyro	200
Accelerometer	74
Radio Altimeter	756
ILS Receiver	252
Air Data System	167
Roll Autopilot Servo	14
Pitch Autopilot Servo	15
Yaw Autopilot Servo	14
EH Valve Drive Coil	1.0
LVDT	.72
Dual Current Comparator (Hardware)	.03
Warning Annunciator (per function)	8.3

*These are NOT actual failure rates for any particular airplane or for any single component produced by a particular manufacturer. They are representative rates determined by a review of generic component types on a number of airplane models in a variety of commercial and military applications. All failure rates per million hours.

9. RELIABILITY PREDICTION USING CARSRA

CARSRA, which stands for Computer-Aided Redundant System Reliability Analysis (Ref. 10), is an analytical reliability prediction program used in the integrated assurance approach to obtain the probabilitty of system failure. In this study, the probability of failure is only considered during the crucial flight phase, which has a duration of 0.02 hours.

The use of CARSRA, along with the quantitative assessment produced by evaluating the fault tree analysis, provides two independent computations of system failure probability. This reduces the risk of a false, low probability of failure being produced by a single method and the error remaining undetected.

Although CARSRA is identified specifically in the integrated assurance approach used in this study, some other method (except fault tree analysis) could be used. If an alternate method is used, it should have sufficient configuration adaptability to produce the predicted probability of system failure without requiring simplifying assumptions which would produce a false, low prediction. Manual analysis is a feasible alternative to CARSRA for many systems.

CARSRA APPLICATION

Configuration Description

Three levels of organization are implicit in the CARSRA inputs, and these levels must be adhered to by the user. At the top level is the system, in this case the RDFCS. System failure probabilities constitute the primary output provided by CARSRA. The intermediate level is comprised of stages. Each stage consists of one or more identical modules, which are at the lowest level. In the RDFCS, each sensor is a module, and like sensors form stages. For example, each of the three normal accelerometers (NA) is a module, and the three NA together comprise a stage.

Markov Models

Markov models were selected by the CARSRA developers as a major part of the program's analytical framework. The following discussion of these models includes some material on applying CARSRA to systems other than the RDFCS. This material is intended to benefit readers not familar with the rationale of developing the input parameters for Markov models as used in CARSRA.

A Markov model is used to describe the number of failed and operating modules within each stage. The transition rates from state to state are used to CARSRA in computing state occupancy probabilities. A separate Markov model is used for each stage. State 1 is the no-failure state in each model, and the two states with the highest numbers correspond to stage failure. The Model always starts in State 1. For example, a dual stage (one of two identical modules required for the stage to function) might have 4 states, as shown in Figure 26. State 1 represents both modules working, State 2 represents one module failed and one working, and States 3 and 4 represent both modules failed. The highest numbered state, 4 in this case, represents undetected stage failure, while State 3 represents detected failure. Note that State 2 does not distinguish which module has failed.

State transition rates must be supplied to CARSRA by the user. These are generally functions of the module failure rates, and possibly other parameters. Returning to the example of the dual stage used previously, the Markov state diagram would be as in Figure 26. Transition rate f_{12} is rate at which transitions occur from State 1 to State 2. That is, if the system is in State 1, the probability that it will transition to State 2 during a short increment of time dt is $f_{12}dt$. The other transition rates are similarly defined.

If there is no monitoring or switching required when the first module fails, and if there is no possibility of the stage failing undetected, the transition from State 1 will always be to State 2, and the transition from State 2 will always be to State 3. Transition rate f_{12} will be simply 2f and f_{23} will be f, where f is the failure rate of a single module. The other transition rates will be 0. Note that this means that State 4 will never be occupied, consistent with undetected stage failure being impossible.

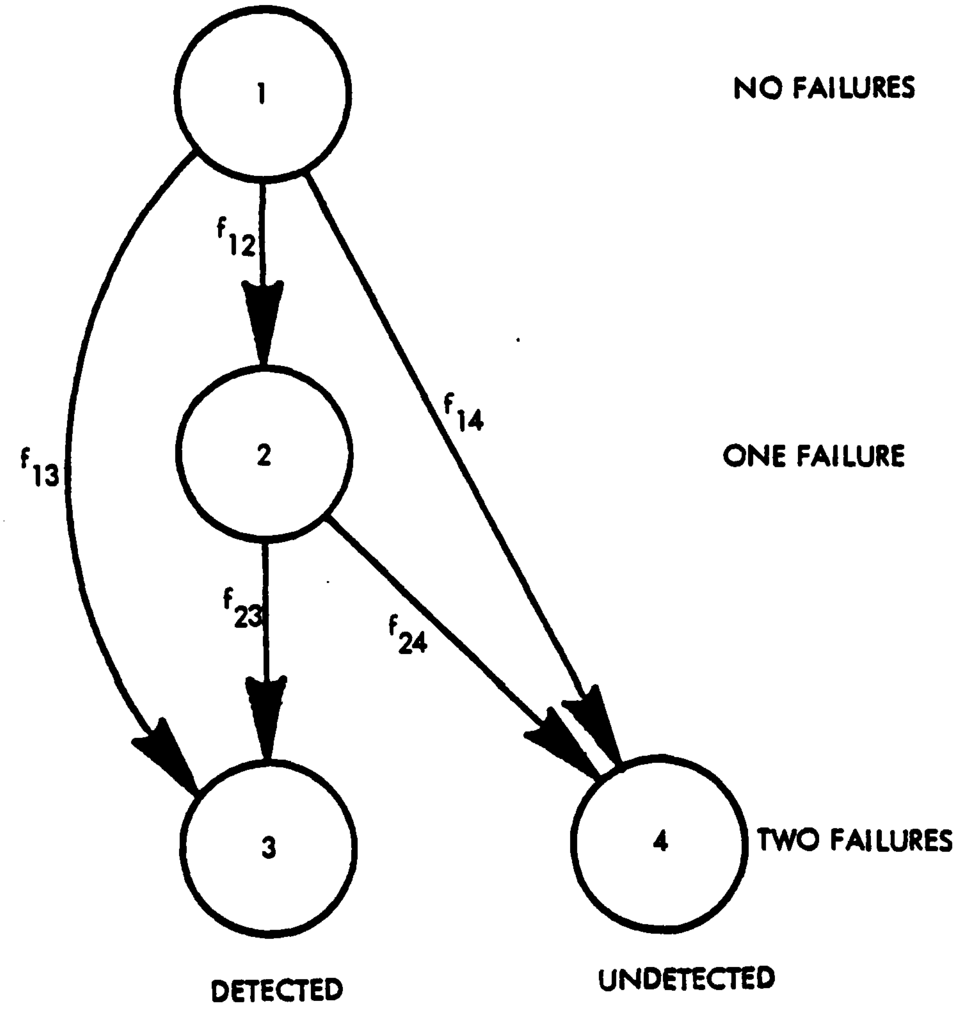


Figure 26. Markov Model of Dual Stage

In many instances encountered in real systems, digital or otherwise, a reconfiguration must occur before the redundancy can be availed. In the example dual case, an output monitor could be used on each module. If the monitor can detect 97% of module failures, e.g. no output or unreasonable output, the monitor provides "coverage", c, of 97%. The transition rate f_{12} is then 2fc, so that 97% of the transitions from State 1 go to State 2.

Of the remaining 3% of the transitions from State 1, some fraction, e.g. 2/3, could go to State 3 and the rest to State 4. This would result in f_{13} being 2f(1-c)(2/3), or 2f(.02), and f_{14} being 2f(1-c) (1/3), or 2f(.01).

Note the distinctions between coverage, which relates to module failure detection, and undetected stage failure. Note also that the function of a particular stage could be such that it cannot fail undetected, even though individual modules within the stage may fail with coverage less than 1. In other cases, stage failure may be detected only by multiple module failures being detected.

It should also be noted that the sum of transition rates out of State 1 is 2f. In general, if any state corresponds to N modules working, the sum of transition rates out of that state will be Nf.

It should be noted also that stages can fail for two reasons, spares exhaustion or coverage failure. In contemporary aircraft systems having critical functions to perform, coverage failures are of as much concern as spares exhaustion.

In the previous dual stage example with 97% coverage of the first module failure, no consideration was included of the failure rate of the monitor itself. The coverage factor of 97% means that 97% of the module faults are of such a nature that they can be detected by an unfailed monitor. The rest are outside of the monitors capability. In cases where dedicated hardware monitors are used, it is appropriate to consider their failure rates and failure modes. A two-state monitor is the type most frequently encountered. It provides only a GOOD/BAD signal. Such a monitor has only two failure states: false indication of BAD when the module is good, and false indication of GOOD when the module is bad.

The simplest way of treating such monitors in CARSRA is to combine the monitors with the modules as a single stage. The transition rate from

State 1 to State 2 is then $2fcr_m + 2f_m a$, where f and c are as before, r_m is the reliability of the monitor over the entire flight time, f_m is the monitor failure rate, and a is the fraction of monitor failures resulting in a good module being declared bad. The other transition rates would be similarly defined, recognizing the relation between detection of stage failure and component monitors. Each instance of such a stage must be evaluated individually in determining the applicable rate formulas.

Frequently, certain terms in a rate equation can be ignored because they are numerically negligible. For example, if $f = 120 \times 10^{-6}$ and $f_m = 0.1 \times 10^{-6}$, the term $2f_m a$ can be ignored in the formula

$$f_{12} = 2fcr_m + 2f_m a,$$

provided c is not absurdly small. If c is 90%, a is 50%, and the flight time is 10 hours,

$$f_{12} = 2(120 \times 10^{-6})(.90) \exp(-.1 \times 10^{-6} \times 10)$$

$$+2(.1 \times 10^{-6})(.50)$$

$$= 216 \times 10^{-6} + .1 \times 10^{-6}.$$

Inclusion of the term yields a rate of 216.1; ignoring it yields 216. The difference is much less than that caused by uncertainty in the module failure rate, 120×10^{-6}.

Dependencies

CARSRA permits the user to describe instances in which failures of a module in one stage will prevent a module in another stage from being used. An example of this in the RDFCS is the portion of each FCC channel which receives sensor data and makes it available to the other channels. Data Acquisition Card A26 in FCC No. 1 receives data from the No. 1 unit of each triple sensor type, and relays it to another card for transmission to the other three channels and for use by its own channel. (Ref. Vol. II,

Section 5.1.1.3.1.5). There are 5 triple-sensor types involved in the autoland mode: pitch, roll, and yaw rate gyros; and lateral and normal accelerometers. (The A26 card also handles data from other sensors, but only these five will be used for discussion here.) If the A26 card fails in FCC No. 1, the data will be lost from pitch gyro No. 1, roll gyro No. 1, yaw rate gyro No. 1, lateral accelerometer No. 1, and normal accelerometer No. 1, just as if all 5 of these sensors had failed. The A26 card is called a dependency module, and its stage a dependency stage. Each of the affected sensors is called a non-dependency module, and the corresponding stage a non-dependency stage.

Coverage for sensor failures is provided by comparison monitoring and reconfiguration (Vol. II, Sec. 5.1.2.4). Each channel independently performs the sensor monitoring functions on the data it will use in control law computations. When a channel detects a failed sensor, it does not tranmit the identity of the individual sensor to the other channels. When a B channel detects a failure, it does transmit a discrete variable, AP.ONEFAIL, to the A channel in the same FCC. The A channel will turn on the NO DUAL annunciation based on its receipt of AP.ONEFAIL from B, or its own detection of a sensor failure. The NO DUAL indication is provided to inform the crew that the RDFCS is not fail-operational. The No. 1 FCC drives the No. 1 Warning Annunciator Indicator (WAI) and the No. 2 FCC drives the No. 2 WAI, so that warning will be provided if either channel of either FCC detects the failure.

The sensor monitoring is part of the foreground flight software. Consequently, for a channel to detect a fault, the CAPS processor must function, as must the CAPS bus and portions of the program and data memory. These are the same hardware elements which perform other functions, such as control law computations and mode logic computatiton. Most faults in these circuit will result in a totally debilitated processor, so that the inability to the monitor sensors is inconsequential. Note also that even if one channel does lose the ability to monitor sensors, any one of the other three channels can force the NO DUAL warning.

In light of the foregoing, the only appreciable probability that the loss of fail-operational sensor capability will not be annunciated results from loss of both WAI. The multiple-function WAI (Ref. Vol. II, Section

5.16.1) has a unit failure rate prediction of 33 per million hours. The failure rate of any one of the 8 warning messages is conservatively taken to be one-fourth the unit rate, or 8.3 per million. It may be noted from Vol. II, Table 5.1.4.6 that the FCC activates the NO DUAL message by providing a ground to the WAI, so that a broken wire or bad connector contact would prevent annunciation. A rate of 1.3 per million hours is included for such failures. Also, the Discrete Output (A20) and Servo Engage Logic (A28) cards are involved, with failure rates of 2.79 and 2.61 per million hours, respectively. Even though only a portion of the failures of these cards will affect NO DUAL, the full rate is used. Further analysis could reduce this rate substantially. The failure rate for NO DUAL is then

WAI	8.3×10^{-6}
Wiring	1.3
A20 Card	2.79
A28 Card	2.61
	15.0×10^{-6}

The probability of failure in a 4-hour time period is then 60×10^{-6}. The Probability of both NO DUAL warnings being lost is the square of this number, 3.6×10^{-9}. It may be noted from Vol. II, Sec. 5.1.2.3.1.1.3 that the test button on the WAI results in the FCC circuitry and the wiring being tested as well as the WAI itself. Thus latent failures are not a problem, provided the indicators are tested prior to autoland.

The factor 3.6×10^{-9} is used as the probability that the first failure of a sensor type will not be covered. This does not constitute stage failure, either detected or undetected. Undetected stage failure is assumed to occur on second failure, provided the first failure was undetected. This is somewhat a misuse of the term "undetected"; the stage failure itself is not necessarily undetected, but the increased likelihood of its occurrence, following first failure, is not annunciated.

This treatment of sensor failures allows the availability feature of CARSRA to be used in computing the probability of loss of one sensor prior to 150 ft., failure of the NO DUAL annunciation, and another failure below 150 ft. The aailability feature is discussed in the next section.

Availability

CARSRA permits system reliability to be computed for a mission phase which follows a period of operation with less stringent failure criteria. An obvious example of this is the RDFCS, which is fail-passive in cruise, but must be fail-operational in autoland below 150 ft. The availability feature allows the user to specify which modules may be failed at the beginning of autoland without forcing diversion to an alternate landing site. Each such availability configuration must provide adequate reliability for the landing, although not as much as if everything is working. The RDFCS requires all of the modules used in autoland to be operational, so that the availability feature might seem not needed in this assessment. It is needed, though, to compensate for a capability which CARSRA lacks.

The reliability of the RDFCS for automatic landing is predicated on the system being fail-operational as the alert height is passed. Therefore, the probability of the system having a latent failure at 150 ft. and a second failure below that point must be quite small.

By setting up the CARSRA input to allow one sensor of each type to fail during cruise, with the transition rate from State 2 to the undetected failure state including the coverage factor of 3.6×10^{-9}, the undetected system failure probability computed by CARSRA will give the probability of an undetected latent failure at 150 ft. and a second failure before touchdown. (See Figure 27)

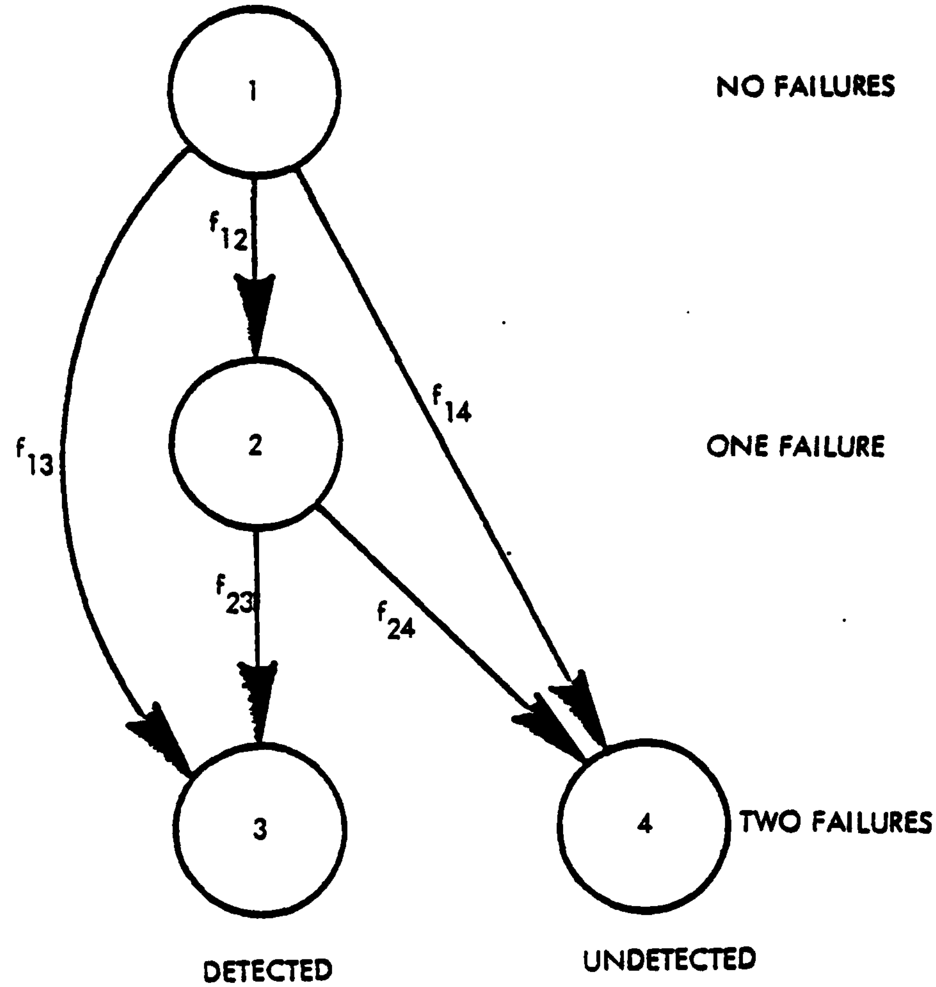
What CARSRA will actually compute is:

P(0 failures at 4 hours) x P(undetected failure and detected failure between 4 and 4.02 hrs.)

+P(1 undetected failure at 4 hours)

x P (second failure between 4 and 4.02 hrs.)

Since the probability of both an undetected and a detected failure between 4 and 4.02 hours is very small, the first term is negligible and



	DUAL SENSOR	TRIPLE SENSOR
f_{12}	2f	3f
f_{13}	0	0
f_{14}	0	0
f_{23}	f	2f
f_{24}	fa	2fa

f = MODULE FAILURE RATE

a = ANNUNCIATION FACTOR 3.6×10^{-9}

Figure 27. Markov Model Coding for Sensor Stages

the output will be equal to the second term, which is the probability desired. This approach is used for the undetected (unannunciated) failures throughout the system. The definition of stages and the transition rates are shown in Figure 28.

The CARSRA program computed some negative probabilities for the unannunciated failures. It is suspected that this may have been caused by the program being run on a Univac 1100-series computer, which has a 36-bit word length. The transition rates to the unannunciated failure states are quite small in some cases (1×10^{-13}), and addition and subtraction of numbers of this magnitude with numbers close to 1.0 could produce some numerical accuracy problems on a 36-bit machine. At NASA-Ames, the program is run on a CDC computer, which has a much larger word size, 64 bits, so that the problem is thought to be unlikely there. Time was not available during the study to investigate and resolve the problem, but this will be done when possible.

Because of the numerical problem encountered with the CARSRA output, the system failure probabilities reported herein were actually manually calculated. This was done by manually computing the stage occupancy probabilities, and then combining these probabilities to account for dependencies between stages, using the same logic that the CARSRA program uses.

The probability of an undetected failure prior to the crucial phase, followed by a second failure in the crucial phase, is 3.36×10^{-14}, compared to 2.46×10^{-14} from the fault trees. The probability of multiple failures in the crucial phase, if everything is working just prior to the phase, is 0.658×10^{-9}, compared with 0.638×10^{-9} from the fault trees.

RDFCS 1/6

```
        12       9
         1       4          2   A500 PWR. SUP
          42.
                 21.
         2       4          2   A400 PWR. SUP.
          42.
                 21.
         3       5          3   ACP 201 PWR. SUP.
          87.
                 58.
                          29.
         4       4          4   A21/221 CARDS
          2.80
                2.10  .0000001
                          1.40
         5       4          4   A26/226 CARDS
          4.80
                3.60  .0000001
                          2.40
         6       4          4   A24/A224 CARDS
          7.20
                5.40  .0000001
```

FIGURE 28. CARSRA INPUT

RDFCS 2/6

			3.60		
7		4	4		A27/227 CARDS
	5.20				
		3.90		.0000001	
			2.60		
8		4	4		A22/23 CARDS
	13.80				
		10.35		.0000001	
			6.90		
9		4	2		PROCESSORS
	116.96				
		58.48		.0000006	
21		4	3		PITCH GYRO
	909.				
		606.		.0000022	
22		4	3		ROLL GYRO
	909.				
		606.		.0000022	
23		4	3		YAW GYRO
	600				
		400.		.0000014	

FIGURE 28. CARSRA INPUT (Cont'd)

RDFCS 3/6

24		4		3	NORM. ACCEL.
	222.				
		148.	.0000005		
25		4		3	LAT. ACCEL.
	222				
		148.	.0000005		
26		4		2	RAD. ALT
	1512.				
		756.	.0000007		
27		4		2	LOC. RCVR
	252.				
		126.	.0000005		
28		4		2	GS. RCVR
	252.				
		126.	.0000005		
29		4		2	DADS
	334.				
		167.	.0000001		
30		4		2	ROLL SERVO

FIGURE 28. CARSRA INPUT (Cont'd)

RDFCS 4/6

```
         28.
                 14.     .0000001
       31         4          2    PITCH SERVO
         30.
                 15.     .0000001
       32         4          2     YAW SERVO
         28.
                 14.     .0000001
    4.0      4.0      .02        .02
       27
  11  41   42   51   52   61   62   71   72   81   82   91
  12  43   44   53   54   63   64   73   74   83   84   92
  21  211  221  231  241  251
  22  212  222  232  242  252
  31  211  221  231  241  251
  32  212  222  232  242  252
  33  213  223  233  243  253
  41  211  221  231  241  251  261  271  281  291
  42  261  271  281  291
  43  212  222  232  242  252  262  272  282  292
  44  262  272  282  292
  51  211  221  231  241  251  261  271  281  291
```

FIGURE 28. CARSRA INPUT (Cont'd)

RDFCS 5/6

52	261	271	281	291					
53	212	222	232	242	252	262	272	282	292
54	262	272	282	292					
61	211	221	231	241	251				
63	212	222	232	242	252				
71	211	221	231	241	251	261	271	281	291
72	261	271	281	291					
73	212	222	232	242	252	262	272	282	292
74	262	272	282	292					
81	261	271	281	291					
82	261	271	281	291					
83	262	272	282	292					
84	262	272	282	292					
91	301	311	321						
92	302	312	322						
	25								
91									
92									
211									
212									
213									
221									
222									
223									
231									

FIGURE 28. CARSRA INPUT (Cont'd)

RDFCS 6/6

1	2	3	4	5	6	7	8	9	10
	2	3	2						
	2	3	3						
	2	4	1						
	2	4	2						
	2	4	3						
	2	5	1						
	2	5	2						
	2	5	3						
	2	6	1						
	2	6	2						
	2	7	1						
	2	7	2						
	2	8	1						
	2	8	2						
	2	9	1						
	2	9	2						
									1
									1

FIGURE 28. CARSRA INPUT (Cont'd)

10. CONCLUSIONS

The conclusions resulting from this study relate to the benefits and limitations of the integrated assurance approach used and the RDFCS Simulator. Certain of the conclusions lead to recommendations, as discussed subsequently.

The primary conclusion drawn from this study is that the integrated assurance approach used is workable for a system, such as the RDFCS, which employs monitoring totally separate from the hardware/software being monitored. In the RDFCS, this monitoring includes the servo coil current comparators and the modulator piston follow-up monitoring. It also includes the warning annunciations which one FCC can generate following a failure in the other FCC. A single-string, self-monitored system might be much less amenable to this approach, depending on the monitoring approaches used. This possibility is outside the scope of this study.

Fault tree analysis is a feasible analytical method for system level faults. One benefit is that specific software failures are identified as the analysis progresses. These can be, and should be, used as a check on the validation test case selection to assure that the software function is rigorously tested. Fault trees can be extended to the circuit card level in a well organized computer such as used in the RDFCS. In general, the analysis is facilitated by a design with clearly partitioned and identifiable functions and interface structure which is consistent for all card inputs and outputs.

Failure mode and effect analysis is more easily accomplished than fault trees within the processor itself. This is because of the processor being involved in a diverse set of functions defined by the flight software. Most individual pin-level faults have many effects. Usually, each fault can be traced to an effect which totally debilitates the processor. Other effects which would also cause massive processor failure, or erroneous results only under certain conditions do not have to be analyzed in detail, provided their effects will not propagate across channels. In contrast, a fault tree analysis based on loss of required system functions would result in identification of the same hardware faults time after time.

The FMEA and fault insertion sessions should be on an iterative basis. After beginning the FMEA, a fault insertion session should be used to confirm the analysis to that point. The results should then be incorporated in the FMEA and the entire FMEA reviewed in light of those results. This review may lead to identification of additional fault cases which should be simulated to resolve uncertainty which may have arisen. This iterative approach was not feasible in this study because of limitations on the availability of the simulator, which was being used on other projects.

The RDFCS simulator has substantial capability for research investigations of digital flight control system validation issues. This capability would be significantly improved by an automated fault insertion and data recording capability. Such a capability should be preprogrammable with a list of faults to be inserted. It should include means of recording the impact of each fault (e.g., changes in the values of discrete variables) for many more variables than the 4 accessible through the CTA's. It should allow variables in channels other than the faulted one to be accessed and recorded.

CARSRA, in its present form, should be used with caution when small failure rates are involved and when execution is to be on a computer with a shorter word length than the 64 bits used in Control Data computers. The possibility of erroneous system failure probability values being output exists under such conditions. This needs to be explored further.

Fault tree analysis and CARSRA provide comparable results for relatively straightforward redundancy conditions, such as the probability of multiple failures during the crucial phase when all components are working at the beginning of the phase. For more complicated situations, the two methods do not agree as closely. This is a result of different simplifications and assumptions being made to structure the problem to the two methods. For example, the third sensor of a triple sensor set (Figure 1) has redundant input paths to the computers (the data input sections of the two computer B channels) but the other sensors have only a single data path (the A channel input sections). This is treated correctly in the fault trees, but the redundancy cannot be accounted for in CARSRA. The conservative assumption is therefore made that loss of either B channel sensor

input capability will cause loss of the third sensor in all triple sensor sets. In validation work, any assumptions required can be made conservatively so that the computed failure probability is actually an upper bound on the true probability.

REFERENCES

1. Federal Aviation Administration, Advisory Circular 25, 1309-1, Subject: Airplane System Design Analysis.

2. RTCA Document DO-178 "Software Considerations in Airborne Systems and Equipment Certification," November 1981.

3. Federal Aviation Administration, Advisory Circular 120-28C, Subject: Criteria for Approval of Category III Landing Weather Minimal, (Draft) May 13, 1982.

4. Lockheed-Georgia Company Engineering Report LG81E0126, "Simulator Investigation Plan for Digital Flight Controls Validation Technology," as revised 10 April 1981.

5. NASA RDFCS System Interface Document, April 8, 1981.

6. Mulcare, D.B. et al: "Industry Perspective on Simulation Methods for Validation and Failure Effects Analysis of Digital Flight Control/Avionics," NASA CR-152234, Moffett Field, California, February 1979.

7. Federal Aviation Administration Report DOT/FAA/CT-82/140, "Digital Flight Control System Validation Technology Assessment," July 1982.

8. Military Standardization Handbook 217C, "Reliability Prediction of Electronic Equipment," United States Air Force, Rome Air Development Center, 9 April 1979, with Notice 1, Supplement, 1 May 1980.

9. Military Specification MIL-M-38510, "Microcircuits, General Specification for."

10. Bjurman, B.E., et al., "Airborne Advanced Reconfigurable Computer System (ARCS)," NASA-CR-145024, Prepared for Langley Research Center, NASA by Boeing Commercial Airplane Company under contract NAS1-13654, August 1976.

APPENDIX A. FMEA RESULTS

Table 3. Pin-Level FMEA

Circuit	Function	Pin	Fault	Effect
Instruction Mapper Prom CU1	Produce direct input bits A0-A3 for control store memory micro-program start address	A0-A9	Open	Address bit low,wrong address read. Wrong output passed to control store proms as starting address bits A0-A3. Massive processor failure.
			Low	Address bit sticks, wrong address read. Wrong output passed to control store proms as starting address. Massive processor failure.
			High	Same as above
		CS1, CS2	Open	Output pins remain in high-impedence state. Input pins to microprogram sequencer CU16 low. Wrong starting address bits A0-A3 to control store proms. Massive processor failure.
			Ground	Don't care.
			High	Same as open.
		01-04	Open	Prom output bit not fed to microprogram sequencer input bit. Input bit low, resulting in wrong microprogram starting address. Massive processor failure.
			Low	Corresponding bit (A0-A3) of microprogram start or jump address is always low. Massive processor failure.
			High	Corresponding bit (A0-A3) of microprogram start or jump address is always high. Massive processor failure.
Instruction Mapper Prom CU7	Produce direct input bits A4-A7 for control store memory microprogram start address.			The fault of any pin of CU7 has the same effect as the same fault occurring in CU1, except that the affected address bits are A4-A7.

Table 3. Pin-Level FMEA (Cont'd.)

Circuit	Function	Pin	Fault	Effect
Instruction Mapper From CU13	Produce direct input bits A8-A9 for control store memory microprogram start address; produce push/pop signals to stack vector register DU4.	A0-A9	Open	Address bit low, wrong address read. Wrong output passed to control store proms as starting address bits A8-A9. Wrong output may also include wrong push or pop signal to stack vector.
			Low	Address bit stuck low, wrong address read. Bits A8-A9 of microprogram start address wrong. Push or pop signal to stack vector register DU4 may be wrong. Massive processor failure.
			High	Address bit stuck high, wrong address read. Bits A8-A9 of microprograms start address wrong. Push or pop signal to stack vector register DU4 may be wrong. Massive processor failure.
		CS1, SS2	Open	Control register cannot pull down enable, so that output pins are at high impedance. Start address bits A8-A9 always low. Massive processor failure.
			Low	Chip CU13 can pull down data input to microprogram sequencer when control register is trying to set it high as part of a jump address.
			High	Same as open.
		01-02	Open	Start address bit A8, A9 to control store always low. Massive processor failure.
			Low	Same as open.
			High	Start address bit A8, A9 high; address bad when bit should be low. Massive processor failure.

Table 3. Pin-Level FMEA (Cont'd.)

Circuit	Function	Pin	Fault	Effect
CU13 con't.		03-04		
			Low	POP commanded (pin 03 faulted) or PUSH commanded (pin 04 faulted) on each clock pulse, so that both commands will go to stack vector register when only non-faulted should be. Stack vector register will do nothing. Massive processor failure.
			High, Open	Fault in pin 03 causes stack vector register to broadside load instead of left shift when mapper prom tries to pop stack. Fault in pin 04 causes broadside load instead of right shift when mapper prom tries to push stack. Stack pointer not pointing to top of stack. Massive processor failure.
Microprogram Sequencer CU14	Generate sequence of microcode Code addresses for control store proms using starting address from mapper or control register. U14 generates microcode address bits A8,A9	D0, D1	Open	Address bit A8, A9 to control store proms always low when starting microcode sequence or on microcode jump. Massive processor failure.
			Gnd.	Same as above
			High	Same as above, except that affected bit is always high.
		CN	Open	Carry-in from microprogram sequencer CU15 is always low. Wrong address will be sent to control store when address increment causes overflow in CU15. Effect depends on allocation of control store store addresses to microcode sequences.
			Gnd.	Same as above

Table 3. Pin-Level FMEA (Cont'd.)

Circuit	Function	Pin	Fault	Effect
CU14 con't.			High	In incrementing address register, bit A8 will be toggled on each clock pulse. Wrong micro-code address will be generated during most microcode sequences. Massive processor failure.
		OE	Open	Address bits A8, A9 always low.
			Ground	No effect during operation. Maintenance trouble-shooting affected.
		RE	Open	Initial microsequence *address from mapper prom* cannot be loaded. Massive processor failure.
			Ground	Same as above.
			High	No effect.
		SO	Open	Sequencer will not jump to proper address when SO should be high. Massive processor failure.
			Ground	Same as above.
			High	Sequencer will execute erroneous jump when SO should be low. Massive processor failure.
		SI		Same effec. as pin SO faulted.
		FE	Open	Microprogram counter will always be pushed onto stack or stack will be popped, depending on PUP. Massive rocessor failure.
			Ground	Same as above.
			High	Microprogram counter cannot be pushed on stack and stack cannot be popped. Massive processor failure.

AD-A129 024 INTEGRATED ASSURANCE ASSESSMENT OF A RECONFIGURABLE DIGITAL FLIGHT CONTROL SYSTEM(U) LOCKHEED-GEORGIA CO MARIETTA W G NESS ET AL. APR 83 DOT/FAA/CT-82/154 2/2

UNCLASSIFIED NAS2-11179 F/G 1/3 NL

END
DATE
FILMED
DTIC

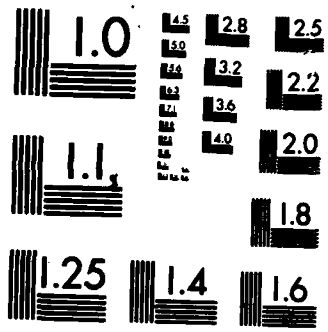

1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0

Table 3. Pin-Level FMEA (Cont'd.)

Circuit	Function	Pin	Fault	Effect
Microprogram Sequencer CU14 con't.		PUP	Open	Stack will be popped when microprogram counter should be pushed on stack. Massive processor failure.
			Ground	Same as above.
			High	Microprogram counter will be pushed onto stack when stack should be popped. Massive processor failure.
		CP	Any	Chip disabled. CP is clock pulse input, and all state changes occur on low-to-high transition of CP. Massive processor failure.
		ZERO	Open	Address bits A8, A9 to control store proms always low. Massive processor failure.
			Ground	Same as above.
			High	Address bits A8, A9 not forced low when commanded by control register CU21. Effect depends on implementation of microcode.
		Y0, Y1	Open	Corresponding address bit to control store is always low. Massive processor failure.
			Ground	Same as above.
			High	Corresponding address bit to control store is always high. Massive processor failure.
		Y2 CN+4, D2,D3	Any	No effect. Pins not connected.
		VCC	Open	Chip dead. Massive processor failure.
		Gnd	Open	Chip dead. Massive processor failure.

Table 3. Pin-Level FMEA (Cont'd.)

Circuit	Function	Pin	Fault	Effect
Microprogram Sequencer CU15	Generate address bits A4-A7 to control store proms. Effects of most pin *faults are the same as* for CU14, except the affected address bits are A4-A7. Only pins with different fault are discussed.	D2,D3	Open	Corresponding address bit A6 or A7 is always when address is from mapper prom or micro-coded jump address. Massive processor failure.
			Gnd	Same as above.
			High	Corresponding address bit A6 or A7 is always high when address is from mapper prom or microcoded jump address. Massive processor failure.
		CN	Open	Carry-in from microprogram sequencer CU14 is always low. Wrong address will be sent to control store when address increment causes overflow in CU14. Massive processor failure.
			Gnd	Same as above.
			High	Carry-in from microprogram sequencer CU14 is always high. Microcode address incremented on bit A4 each clock cycle. Massive processor failure.
		CN+4	Open	Same as CN open on CU14.
			Gnd	Same as above.
			High	Same as CN high on CU14.
		Y2,Y3	Open	Corresponding bit A6 or A7 always low in address to control store. Massive processor failure.
			Gnd	Same as above.
			High	Corresponding bit A6 or A7 always high in address to control store. Massive processor failure.

Table 3. Pin-Level FMEA (Cont'd.)

Circuit	Function	Pin	Fault	Effect
Microprogram Sequencer CU16	Generate address bits A0-A3 to control store proms. Effects of most pin faults are the same as for CU15, except that affected address bits are A0-A3. Only pins with different effects are discussed.	CN	Open	Microprogram address is not incremented during execution of microcode sequence. Massive processor failure.
			Gnd	Same as above.
Micro-processor DU17	Processes the four low-order bits of the 16-bit CAPS word in response to instructions from control registers.	A0-A3	Open	Wrong A pointer address when failed bit should be high. Massive processor failure.
			Gnd.	Same as above.
			High	Wrong A pointer address when failed bit should be low. Massive processor failure.
		B0-B3	Open	Wrong B pointer address when failed bit should be high. Massive processor failure.
			Gnd	Same as above.
			High	Wrong B pointer address when failed bit should be low. Massive processor failure.
		I0-I2	Open	Wrong data source selected when failed bit should be high. Massive processor failure.
			Gnd	Same as above.
			High	Wrong data source selected when failed bit should be low. Massive processor failure.

Table 3. Pin-Level FMEA (Cont'd.)

Circuit	Function	Pin	Fault	Effect
Micropro-cessor DU17		13-15	Open	Wrong operation performed when failed bit should be high. Massive processor failure.
			Gnd	Same as above.
			High	Wrong operation performed when failed bit should be low. Massive processor failure.
		16-18	Open	Wrong destination code when failed bit should be high. In most cases, the immediate effect will be internal to the chip involving load or shift of data in registers. Massive processor failure.
			Gnd.	Same as above.
			High	Wrong destination code when failed bit should be low. Massive processor failure.
		CP	Any	Chip dead. Massive processor failure.
		D0-D3	Open	Input to processor is wrong when failed bit should be high. Major effect caused by incorrect bit in packed Boolean data. Massive processor failure.
			Gnd.	Same as above.
		C	Open	Carry-in always low. Program counter not incremented on instruction fetch. Massive processor failure.
			Gnd.	Same as above.
			High	Carry-in always high. Foreground loop of flight software cannot execute paths 2 and 4; iteration monitor test bit not toggled; iteration monitor trips. FCC disconnects.

Table 3. Pin-Level FMEA (Cont'd.)

Circuit	Function	Pin	Fault	Effect
Micropro-cessor DU17 con't.		Y0-Y3	Open	Wrong address gated on CAPS address lines. Massive processor failure.
		P	Open	Carry propagate always sent to carry look-ahead logic. Massive processor failure.
			Gnd.	Same as above.
			High	Carry propagate never sent to carry look-ahead logic. Double-precision integrators drift.
		G	Open	Carry generate always sent to carry look-ahead logic. Massive processor failure.
			Gnd.	Same as above.
			High	Carry generate signal never sent to carry look-ahead logic.
		F=0	Open	DU17 cannot pull down F=0 line to status register, yielding false results for some logic tests. Massive processor failure.
			Gnd.	DU17 always pulls down F=0 line to status register yielding false results for some logic tests. Massive processor failure.
			High	Same as open.
		Vcc	Open	Chip dead. Massive processor failure.
		OE	Open	Chip dead. Massive processor failure.
		Gnd.	Open	Chip dead. Massive processor failure.
		R3	Open	Bit left-shifted into DU14 or right shifted into DU17 always low. Multiplication results erroneous. VCC disconnect.
			Gnd.	Same as above.

Table 3. Pin-Level FMEA (Cont'd.)

Circuit	Function	Pin	Fault	Effect
Microprocessor DU17 cont't.			High	Bit left-shifted into DU14 or right-shifted into DU17 always high. Multiplication results erroneous. FCC disconnect.
		Ro	Open	Bit right-shifted to shift/rotate multiplexer or input from shift/rotate multiplexer always low. Multiplication results erroneous. FCC disconnects.
			Gnd.	Same as above.
		RO	High	Bit right-shifted to shift/rotate multiplexer or input from shift/rotate multiplexer always high.
		Q3	Open	Bit right-shifted into DU17 or left-shifted to DU14 always low. Multiplication results erroneous. FCC disconnects.
			Gnd.	Same as above.
			High	Bit right-shifted into DU17 or left-shifted to DU14 always high. Multiplication results erroneous. FCC disconnects.
		Q0	Open	Bit right-shifted to shift/rotate multiplexer or left shifted from shift/rotate multiplexer always low. Multiplication results erroneous. FCC disconnects.
			Gnd.	Bit right-shifted to shift/rotate multiplexer. or left-shifted from shift/rotate multiplexer always high. Multiplication results erroneous. FCC disconnects.
		F3	Any	No effect. Pin not connected.
		CN+4	Any	No effect. Pin not connected.
		OVR	Any	No effect. Pin not connected.

Table 3. Pin-Level FMEA (Cont'd.)

Circuit	Function	Pin	Fault	Effect
Micro-processor DU14	Microprocessor DU14 handles bits 4-7 of the 16-bit APS word in response to instructions from the control registers. Effect of fault as if fault had occurred on DU17, except that different bit positions in the CAPS word are affected.			
Micro-processor DU18	Microprocessor DU18 handles bits 8-11 of the 16-bit CAPS word in response to instructions from the control registers. Effect of pin faults is the same as if the fault had occurred on DU17, except that different bit positions in the CAPS word are affected.			
Micro-processor DU15	Microprocessor DU15 handles bits 12 - 15 of the 16-bit CAPS word in response to instructions from the control registers. Effect of pin faults is the same as if the fault had occurred in DU14, except that different bits are affected. Some differences result from the use of bit 15 as the sign bit in numerical computation.			

APPENDIX B. FAULT SIMULATION RESULTS

Table 4. Faults Simulated

COMPONENT LEVEL FAULTS

CASE	ADDRESS VARIABLE	FB03 VG #1	FB01 $\dot{\theta}$ #1	3635 AP. ONE FAIL	3318 EXEC FAIL θ	FB03 VG #3	FB01 $\dot{\theta}$ #3	32E6 EXEC FAIL θ	331A EXEC FAIL $\dot{\theta}$	
	Strip Chnl	1	2			3				
1A	Vert. Gyro #1 X-Leg open inbound			0000 0001	0000 0003			0000 0003	0000 0003	Pin PIA-31, X-Leg V.G. #1 Open On Pin removal; ATS Disconnect warning - NO DUAL, NO ALIGN at 1500 ft.
1B	Vert. Gyro #1 Y-Leg open inbound			0000 0000 0000	0000 0003 0000			0000 0003 0000	0000 0000 0000	On pin removal ATS disconnect; prior to A/L track AP.ONEFAIL was cleared.
2A	Vert. Gyro #1 hard shift to fixed value inbound			0000 0001	0000 0003			0000 0003	0000 0000	First fault was not of sufficient magnitude to fault sys., fault level was increased resulting in the CTA values shown, NO DUAL was annunciated upon engagement of A/L TRK.
2B	Same as 2A but in Al. Arm			0000 0001 0000	0000 0003 0003			0000 0003 0003	0000 0000 0000	ATS Disc. WRN. on insertion NO DUAL did not light. AP.ONEFAIL reset.
2C	Same as 2A but in AL. TRK above 150 ft.			0000 0000	0000 0003			0000 0003	0000 0000	ATS. dropped out on insertion; no ATS warning; NO DUAL did not annunciate
2D	Same as 2A but in AL. TRK below 150 ft.			0000 0000	0000 0003			0000 0003	0000 0000	NO DUAL did not annunciate
2E	Vert. Gyro #1 Validity false inbound			0000 0001	0000 0000			0000 0000	0000 0000	NO DUAL did not annunciate

Table 4. Faults Simulated (Cont.)

COMPONENT LEVEL FAULTS

CASE	ADDRESS / VARIABLE	FB03 VG #1	FB01 $\dot{\theta}$ #1	3635 AP. ONE FAIL	3318 EXEC FAIL θ	FB03 VG #3	FB01 $\dot{\theta}$ #3	32E6 EXEC FAIL θ	331A EXEC FAIL $\dot{\theta}$	
	Strip Chnl									
3A	Vert. Gyro #1			0000	0000			0000	0000	Disconnect of ATS on insertion;
	Open X-Leg in			1	0000			0000	0000	No dual annunciated.
	AL.ARM									
3B	Vert. Gyro #1			0000	0000			0000	0000	ATS Disc. on insertion.
	Open Y-Leg in			1	3			3	3	No dual annunciated.
	AL.ARM			0	3			3	0	Exec. Fail Resets
4A	Vert. Gyro #1			0000	0000			0000	0000	No dual annunciation, ATS
	Open X-Leg in			1	3			3	3	Disc. Annun. ATS Disc.
	AL.TRK above									
	150 Ft.									
4B	Vert. Gyro #1			0000	0000			0000	0000	No dual & ATS Disc.annunciated,
	Open Y-Leg			1	3			3	3	ATS Disc.
	AL.TRK above									
	150 Ft.									
5A	Same as 4A but			0000	0000			0000	0000	ATS Disconn. without annunciation.
	below 150 Ft.			1	3			3	3	No dual not indicated.
5B	Same as 4B but			0000	0000			0000	0000	No dual did not indicate; ATS
	below 150 Ft.			0	3			3	0	Disc. & Ind.

Table 4. Faults Simulated (Cont.)

COMPONENT LEVEL FAULTS

CASE	ADDRESS VARIABLE	FB03 VG #1	FB01 $\dot{\theta}$ #1	3635 AP. ONE FAIL	3318 EXEC FAIL θ	FB03 VG #3	FB01 $\dot{\theta}$ #3	32E6 EXEC FAIL θ	331A EXEC FAIL $\dot{\theta}$	
	Strip Chnl									
6	Vert. Gyro #1			0000	0000			0000	0000	No dual flashed but did not
	Ramp Up			1	3			3	0	latch.
	Inbound			0	3			3	0	3635 reset when no dual flashed.
7	Vert. Gyro #1			0000	0000			0000	0000	No dual indicated.
	Ramp Down			1	3			3	0	
	Inbound									
8A	Vert. Gyro #3			0000	0000			0000	0000	
	Open X-Leg			1	2			2	0	
	Inbound			0	2			2	0	At 1500 Ft. No dual lit
8B	Vert. Gyro #3			0000	0000			0000	0000	No dual flashed, Reset 3635
	Open Y-Leg			1	2			2	0	ATS Disc. Warning but ATS stayed.
	Inbound			0	2			2	0	

Table 4. Faults Simulated (Cont.)

COMPONENT LEVEL FAULTS

CASE	ADDRESS VARIABLE	FBIF N.A.#1	3380 N.A.EXEC. FAIL	3635 ARONE FAIL	FBIF N.A.#3	334E EXEC. FAIL	
	Strip Chnl	1			3		
9A	Norm. Accel. #1 Open signal Inbound		0000 0	0000 0		0000 0	ATS held in; ATS Disc. warning at A/L TRK.
			0000 3	0000 1		0000 3	Case 9A repeated with turbulence; Fault detected & NO DUAL
9B	Norm. Accel. #1 Open Gnd. Inbound		0000	0000		0000	Case 9B involved signal gnd. Pulling pin has no effect.

Table 4. Faults Simulated (Cont.)

COMPONENT LEVEL FAULTS

FCC # 1 CTA WINDOW

CASE	ADDRESS VARIABLE STRIP CHNL	A1 FB07 ROLL RATE 1	A2 FB05 ROLL RATE 1	A3 3635 A.P.ONE FAIL	A4 33E8 EXEC. FAIL ROLL	B1 FB07 ROLL RATE 3	B2 FB05 ROLL RATE 3	B3 3386 EXEC. FAIL ROLL	B4 33EA EXEC. FAIL ROLL	
10A	Roll Gyro #1 Open X-Leg in AL.ARM			0000 1	0000 3			0000 3	0000 3	No dual at A/L TRK
10B	Roll Gyro #1 Open Y-Leg in AL.ARM			0000 1	0000 •3			0000 3	0000 0	No dual at A/L TRK
11A	Roll Gyro #1 Ramp Up Inbound			0000	0000			0000	0000	No dual at A/L TRK
11B	Same as 11A but in AL.ARM			0000 1	0000 3			0000 3	0000 0	No dual at 1100 FT. (When Comparators Tripped).
11C	Same as 11A			0000 1	0000 3			0000 3	0000 0	No dual at 1100 FT. (When Comparators Tripped).
11D	Same as 11A but in AL.TRK below 150 Ft.			0000 1	0000 3			0000 3	0000 0	Landing Completed Without No Dual Indication.

Table 4. Faults Simulated (Cont.)

COMPONENT LEVEL FAULTS

FCC # 1 CTA WINDOW

CASE	ADDRESS VARIABLE STRIP CHNL	A1 FB1B LOC.DEV. # 1	A2 339A EXEC.FAIL LOC.DEV.	A3	A4 33E8	B1 FB1B LOC.DEV. # 1	B2 3368	B3	B4 33EA	
12A	No Localizer Output In Inbound		0000 6003 4003		0000 0		0000 6003 4003		0000 0	No dual at AL.TRK
12B	Same as 12A but in AL.ARM		0000 4003		0000 0000		0000 4003		0000 0000	No dual indicated when fault detected (at 1000 Ft.)
12C	Same as 12A but in AL.TRK above 150 Ft.		0000 4003		0000 0000		0000 4003		0000 0000	No dual indicated when fault detected (at 1000 Ft.)
12D	Same as 12A but in AL.TRK below 150 Ft.		0000 6003		0000 0000		0000 6003		0000 0000	Fault detected; Land completed; No dual did not illuminate.
14A	Localizer No. 1, Validity 1 in AL. ARM		0000 4013		0000		0000 4013		0000	No dual at AL.TRK
14B	Localizer No. 1, Validity 2 in AL. ARM		0000 4023		0000		0000 4023		0000	No dual at AL.TRK
16	Lateral Accel. No. 1 Ramp Up in AL. ARM	FB1D	33B4	33B4 EXEC.FAIL 0000 400A	3635 AP. ONE FAIL 0000	FB1D		3382 EXEC.FAIL 0000 400A	3635 AP.ONE FAIL 0000	No dual at 1100 Ft. when comparator tripped.

Table 4. Faults Simulated (Cont.)

COMPONENT LEVEL FAULTS

FCC #1 CTA WINDOW

CASE	ADDRESS VARIABLE	A1	A2	A3 3364 AP.TWO FAIL	A4 3365 AP.ONE FAIL	B1	B2	B3 3364 AP.TWO FAIL	B4 3365 AP.ONE FAIL	
	STRIP CMD.									
17A	Pitch Servo Coil Discrete Fault Inbound			0000	0000			0000	0000	Instant disengage; No dual at A/L TRK. Servo simulator panel pitch coil switch to fault. Box 2 engaged first.
17B	Same as 17A but in AL.ARM			0000 0000	0000 0000			0000 0000	0000 0000	Both bathandles dropped.
17B	Rerun									On 3rd retry, only affected channel disengaged, no dual at A/L TRK.
17C	Same as 17A, but in AL.TRK above 150 Ft.									Affected FCC disengaged; No dual
17D	Same as 17A, but in AL.TRK below 150 Ft.									Affected FCC disengaged; No dual not indicated.
18A	Pitch Servo Coil Current Ramp Up Inbound									Affected channel disengaged; No dual at A/L.TRK
19A	Roll Servo Discrete Open Inbound									
19B	Same as 19A but in AL.ARM									Affected bathandle dropped; No dual indicated.
19C	Same as 19A but in AL.TRK above 150 Ft.									Affected bathandle dropped; No dual indicated.
19D	Same as 19A but in AL.TRK below 150 Ft.									Affected bathandle dropped; No dual not indicated.

Table 4. Faults Simulated (Cont.)

Case	Circuit	Pin No.	Pin Function	Open Pin Faults
20a	DU18 2901 No.3 (data bits 8-11)	24	Data Input Bit 1	Upon opening, both FCC's disengaged. NORMAL-STANDBY switch was in STANDBY.
20		11	F = 0	Faulted FCC disengaged. SPLIT and NO DUAL annunciated. Faulted pin transmits signal to the status register.
20b		4	Address bit A_0	Faulted FCC disengaged. SPLIT, NO DUAL, and NO ALIGN annunciated at initiation of AL TRK.
20c		17	Address bit B_0	Both FCC's disengaged. AP DISC and SPLIT annunciated.
20d		37	Data Output bit 1	Faulted FCC disengaged. AP DISC and SPLIT annunciated on disengagement. NO DUAL annunciated at AL TRK.
21	CU36	1		Fault had no effect on computer operation. Pin used only in reset of stack vector and transfer bus access control registers.
22a	CU30 Control Register 29LS18	1	Data Input bit 0	Faulted FCC disengaged. NO DUAL, NO ALIGN, and SPLIT annunciated. Pin 1 is coupled to Pin 2 (CR35, see fault following) and, when the next address control prom sets OE low, to direct input bit D1 of microprogram sequencer CU14.
22b		2	Data Output bit Q_0	Faulted FCC disengaged. SPLIT, NO DUAL annunciated. Faulted bit drives control line CR35, which is address bit A3 of the 2901's when the processor address multiplexer couples CR35 to address line A03.
22c		10	Data Output bit Y2	Faulted FCC disengaged. SPLIT, NO DUAL, and NO ALIGN annunciated. Faulted pin is direct input bit D3 to microprogram sequencer CU15 when bit D3 is not being controlled by instruction mapper prom CU7. In turn, CU15 outputs this bit as address bit A7 to the control store proms when CU1E is in direct address mode.

Table 4. Faults Simulated (Cont.)

Case	Circuit	Pin No.	Pin Function	Open Pin Faults
22e	CU30	4	Data Input bit D1	Faulted FCC disengaged; other FCC went to CWS. CMD DISC annunciated. Control line CR34 is latched to bit D1 on rising clock pulse, and when next address control prom sets OE low, to direct input bit D0 of microprogram sequencer CU14. When selected by data select multiplexer DU28, CR34 is used as data bit D06.
22c		4		In a repeat of previous case, faulted FCC disengaged. Other FCC stayed in CMD. SPLIT and NO DUAL annunciated.
22f		12	Data Input bit D2	Faulted FCC disengaged, other FCC stayed in CMD. NO DUAL and SPLIT annunciated. Control line CR33 is latched to Pin 12 on rising clock pulse. CR33 is used as processor A address bit AP1 when connected by the A address multiplexer. Also, CR33 can be coupled to processor input data bit D05 by data select multiplexer DU21.
23a	CU15 Micro-program Sequencer 2911	9	Not Zero	Faulted FCC disengaged. SPLIT and NO DUAL annunciated at ALIGN point in landing. Pin 9 forces all outputs of CU15 to zero when it is low. Open pin prevented Hi signal from control register CU21 from reaching CU15, zeroing all outputs and causing erroneous address to control store memory.
23b		19	Not FE	Faulted FCC disengaged. Processor halted. System failed to capture glide slope. The FE signal is one of four used to control the operation of the 2911 micro-program sequencer. In most combinations of the signals, the absence of the FE signal causes a push or pop of a counter stack in addition to a jump.
23c		19	Not FE	Repeat of previous fault. Faulted FCC disengaged. Other FCC stayed in CMD. SPLIT and NO DUAL annunciated.
23d		20	PUP	Faulted FCC disengaged. NO DUAL and SPLIT annunciated. PUP is Push/Pop control signal. Open pin prevents pushing the microprogram counter contents onto the internal stack.

Table 4. Faults Simulated (Cont.)

Case	Circuit	Pin No.	Pin Function	Open Pin Faults
23e	CU15 Microprogram Sequencer 2911	10	SO Address Source Selection Control	Faulted FCC disengaged. NO DUAL and SPLIT annunciated. SO is one of the four signals used in selecting the source of the next address. SO open generally results in wrong source producing a jump to the wrong address.
23f		5	D2 Direct Input bit D2	Faulted FCC disengaged. NO DUAL and SPLIT annunciated. D2 is one of four bits which can be selected as the output of the 2911. Fault would cause the wrong control store memory address on selection of direct input when the bit should be high.
23g		15	Y3 Output bit	Faulted FCC disengaged. NO DUAL and SPLIT annunciated. Y3 is one of four output bits of the 2911. This bit being open causes the wrong microinstruction to be selected whenever this bit should be high.

Table 4. Faults Simulated (Cont.)

Case	Circuit	Pin No.	Pin Function	*Grounded Pin Faults*
24	CU2 Hex Inverter	8	Inverter Output	Faulted FCC disengaged. Processor stopped. This signal fans out to several points, including NAND gate CU35C, which outputs the Read Enable signal to the data bus transceivers. NAND output is stuck high so that processor cannot read the CAPS data bus.
25	CU36 Quad. NOR Gate	4	Gate Output	Faulted FCC disengaged. NO ALIGN, SPLIT, and NO DUAL annunciated. Faulted pin being stuck low results in NAND gate U35A being stuck high, disabling the data bus transceivers from writing on the CAPS data bus.
26	CU28 Quad. NAND Gate	3	Gate Output	Faulted FCC disengaged. NO ALIGN, SPLIT, and NO DUAL annunciated. Processor stopped. Fault results in BROF being stuck high, so that processor cannot access CAPS bus. Also, all interrupt inputs to the interrupt controller are set high.
27	CU36 Quad. NOR Gate	13	Gate Output	Faulted FCC disengaged. NO ALIGN, SPLIT, and NO DUAL annunciated. Processor stopped. Fault results in the processor being unable to transmit XAKF (transfer acknowledge) on the CAPS control bus. XAKF is stuck high.
28	DU2 Shift/ Rotate Multi-plexer	2	Control Input	Faulted FCC disengaged. SPLIT, NO ALIGN, NO DUAL annunciated. Faulted processor stopped. Fault causes wrong data to be inserted into microprocessor in some shift operations.
		14	Control Input	Faulted FCC disengaged. Fault causes wrong data to be inserted into microprocessors during some shift operations.

Table 4. Faults Simulated (Cont.)

Grounded Pin Faults

Case	Circuit	Pin No.	Pin Function	
29	DU17 Micro-processor	36	Y00	
		37	Y01	
		38	Y02	
		39	Y03	
30	DU14 Micro-processor	36	Y04	In each case, the faulted FCC disengaged. The faulted processor halted immediately. The y pins are the processor output pins for computed data. Under certain conditions, processor output is a memory address which is connected to the CAPS address bus, rather than data. Corruption of addresses is apparently the cause of the immediate processor halts.
		37	Y05	
		38	Y06	
		39	Y07	
31	DU18 Micro-processor	36	Y08	
		37	Y09	
		38	Y10	
		39	Y11	
32	DU15 Micro-processor	36	Y12	
		37	Y13	
		38	Y14	
		39	Y15	

Table 4. Faults Simulated (Cont.)

Case	Circuit	Pin No.	Pin Function	Grounded Pin Faults
33	DU17 Micro-processor	32	$\bar{C}$	
		35	$\bar{P}$	
34	DU14 Micro-processor	32	$\bar{C}$	Faulted FCC disengaged. Faulted processor halted immediately.
		35	$\bar{P}$	
35	DU18 Micro-processor	35	$\bar{P}$	
36	DU16 Interrupt Controller 2914	16	V2	Faulted FCC disengaged. Faulted processor stopped immediately. V2 is the most significant bit of the interrupt vector output of the 2914. This bit is also address bit A02 of the CAPS address bus when the vector output is enabled, and is hard-wired to address line A02.
37		17	V1	Faulted FCC disengaged. Faulted processor stopped immediately. V1 is the middle bit of the three-bit interrupt vector of the 2914. This pin is hard-wired to CAPS address bus line A01.
38		18	V0	Faulted FCC disengage. Faulted processor stopped immediately. V0 is the least significant bit of the interrupt vector output of the 2914. This pin is hard-wired to CAPS address bus line A00.
39		28	I0	Faulted FCC disengaged. Faulted processor stopped immediately. I0 is a micro-instruction bit to the 2914.
40		31	I1	Faulted FCC disengaged. Faulted processor stopped immediately. I1 is a micro-instruction bit to 2914.
41		32	I2	Faulted FCC disengaged. Faulted processor stopped immediately. I2 is a micro-instruction bit to the 2914.

Table 4. Faults Simulated (Cont.)

GROUNDED PIN FAULTS

Case	Circuit	Pin No.	Function	
42a	DU16 Interrupt Controller 2914	34	Instruction Enable	Faulted FCC disengaged. Faulted processor halted. Pin 34 is a logic-low instruction enable which should only go low when the instruction lines I0-I3 have been set. The pin stuck low causes the 2914 to read erroneous instructions.
42b		26	P4 Interrupt Request	Faulted FCC disengaged. Faulted processor halted. Pin 34 is a logic-low interrupt request. With the fault inserted, an interrupt request at priority 4 is generated whenever the corresponding mask bit is not set and a higher priority unmasked interrupt is not present.
42c		39	P2 Interrupt Request	Faulted FCC disengaged. Faulted processor halted. This is the same situation as in the previous case, except at a lower priority level.
42d		20	P7 Interrupt Request	Faulted FCC disengaged. Faulted processor halted. This is the same situation as in the previous two cases, except at the highest priority level.
42e		25	M4 Mask Bit	Faulted FCC disengaged. Faulted processor halted. This fault prevents priority Level 4 interrupts from being masked.
42f		19	M7 Mask Bit	Faulted FCC disengaged. Faulted processor halted. This fault prevents highest priority interrupts from being masked.

Table 4. Faults Simulated (Cont.)

CASE	ADDRESS VARIABLE	A1	A2	A3	A4	B1	B2	B3	B4	
				FCC #1 CTA WINDOW						
	ADDRESS VARIABLE			AP. ONE FAIL	AP.TWO FAIL					
	STRIP CHNL			3635	3634					
43	Open Roll Gyros 1 & 2 in AL.ARM			0000 0001	0000 0001 0000					Disconnected on Second Fault. 3634 flashed 0001 before reverting to 0000.
44	Open roll Gyros 1 & 3 to FCC #1 in AL.ARM									Sensors 2 & 3 still valid into box 2. No disconnect; No dual at AL.TRK.
45	Ramp Up Vert. Gyro #1 in APP; Open Vert. Gyro #2 in AL.ARM									Two sensors lost; both boxes disengaged.

APPENDIX C. PROCESSOR SCHEMATIC DIAGRAMS

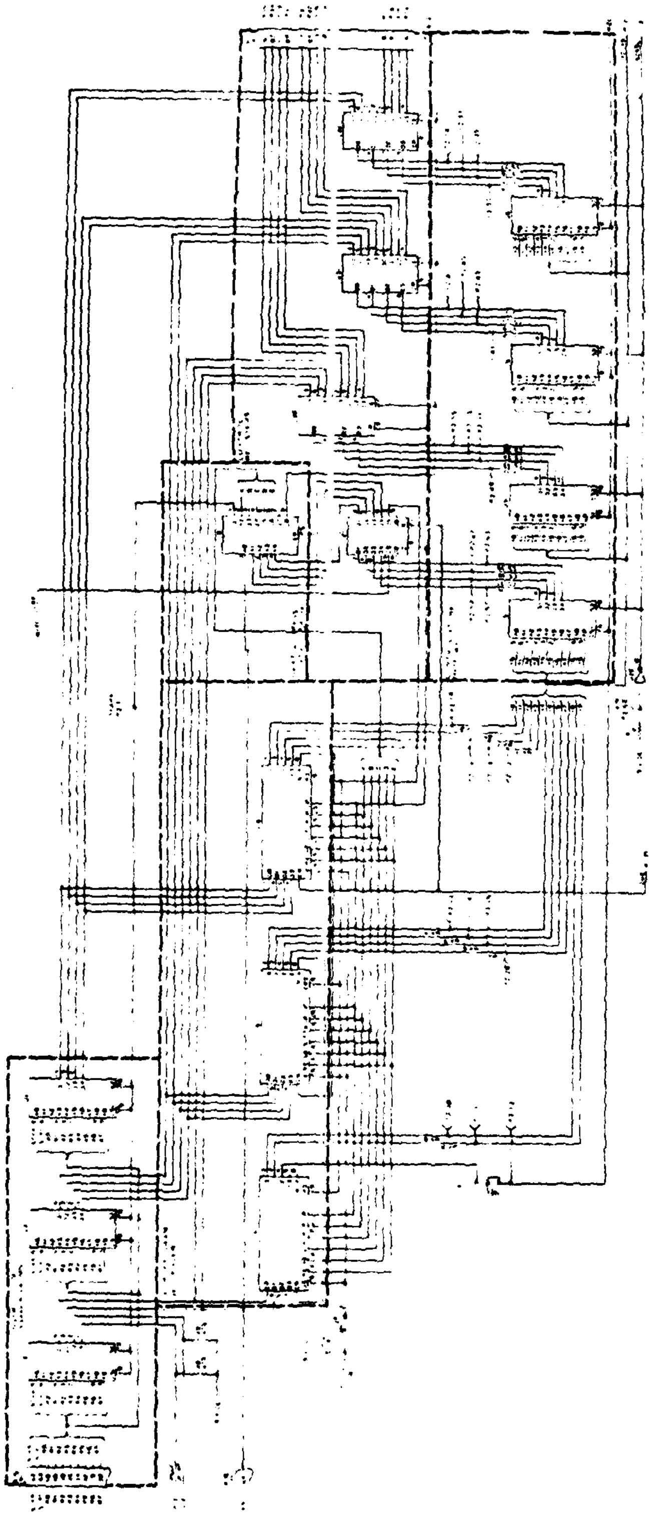


FIGURE C-1. CONTROL CARD SCHEMATIC DIAGRAM (SHEET 1 of 3)

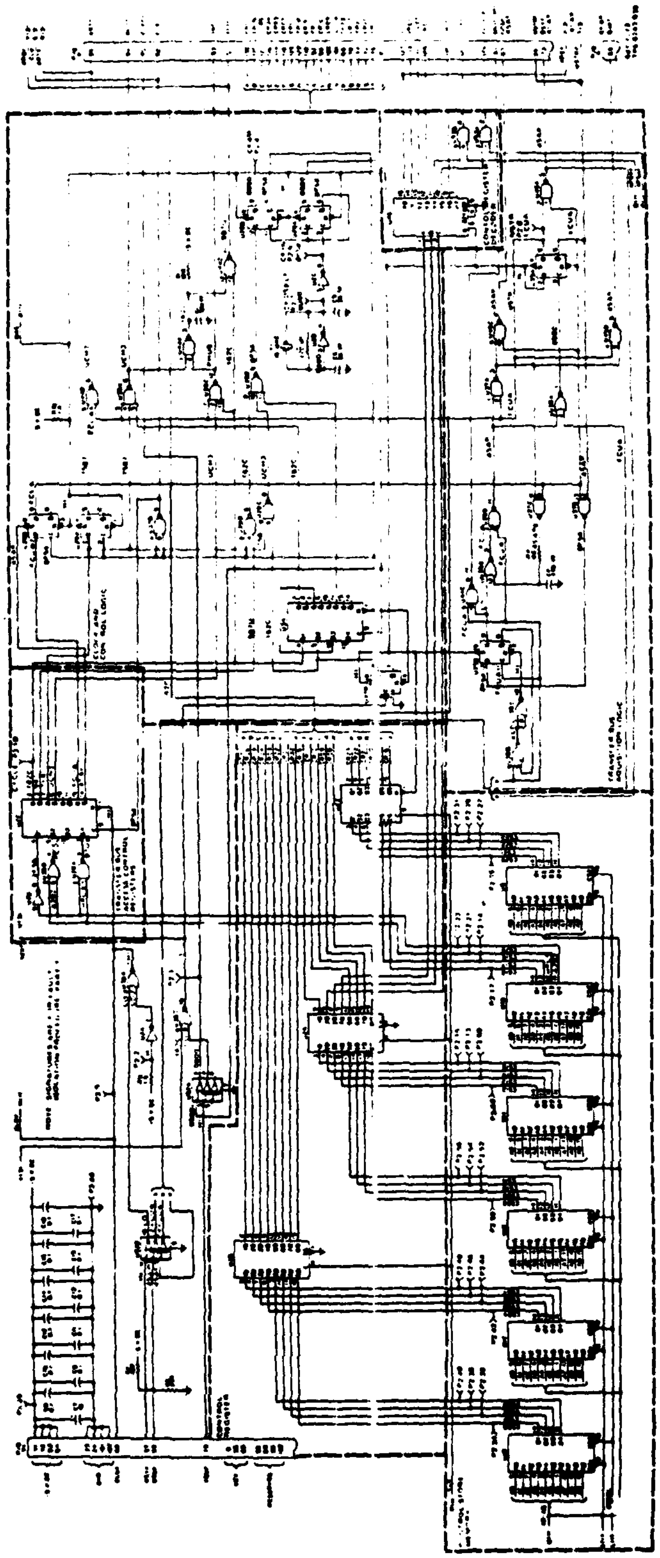


FIGURE C-1. CONTROL CARD SCHEMATIC DIAGRAM (SHEET 2 of 3)

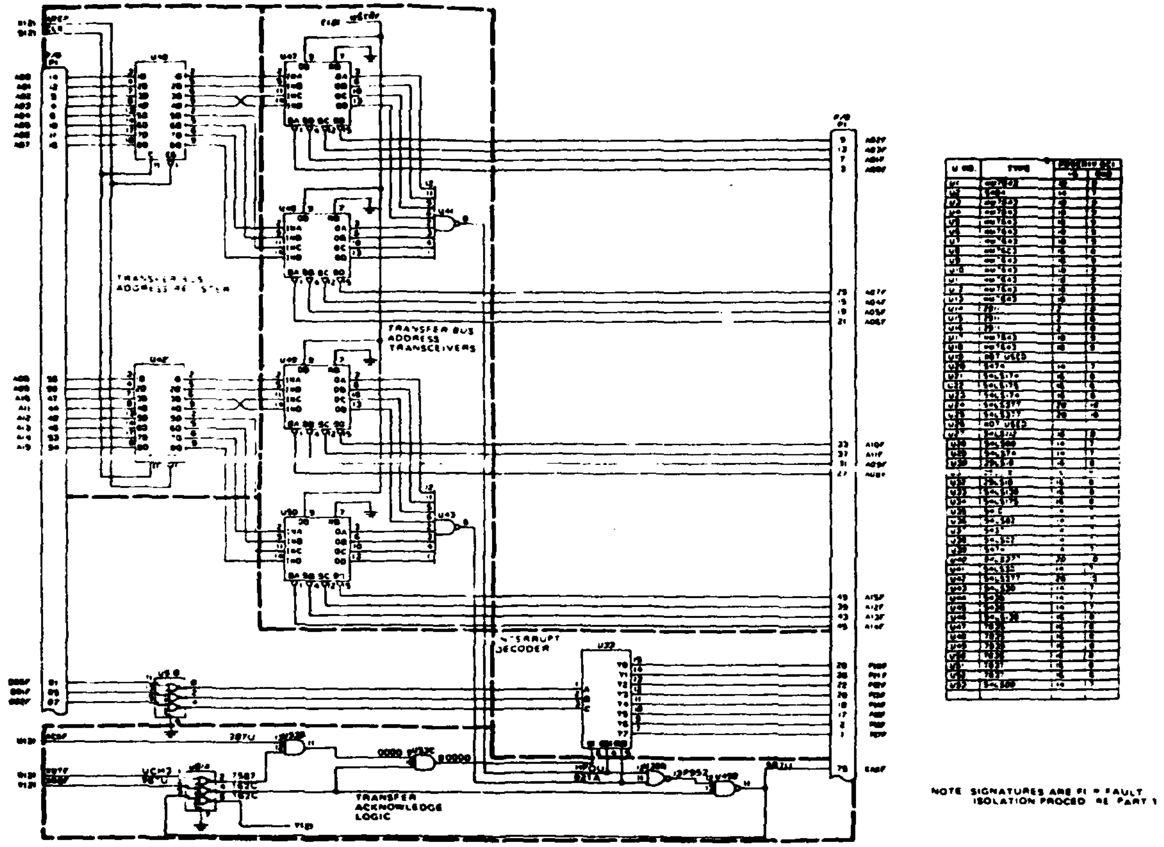
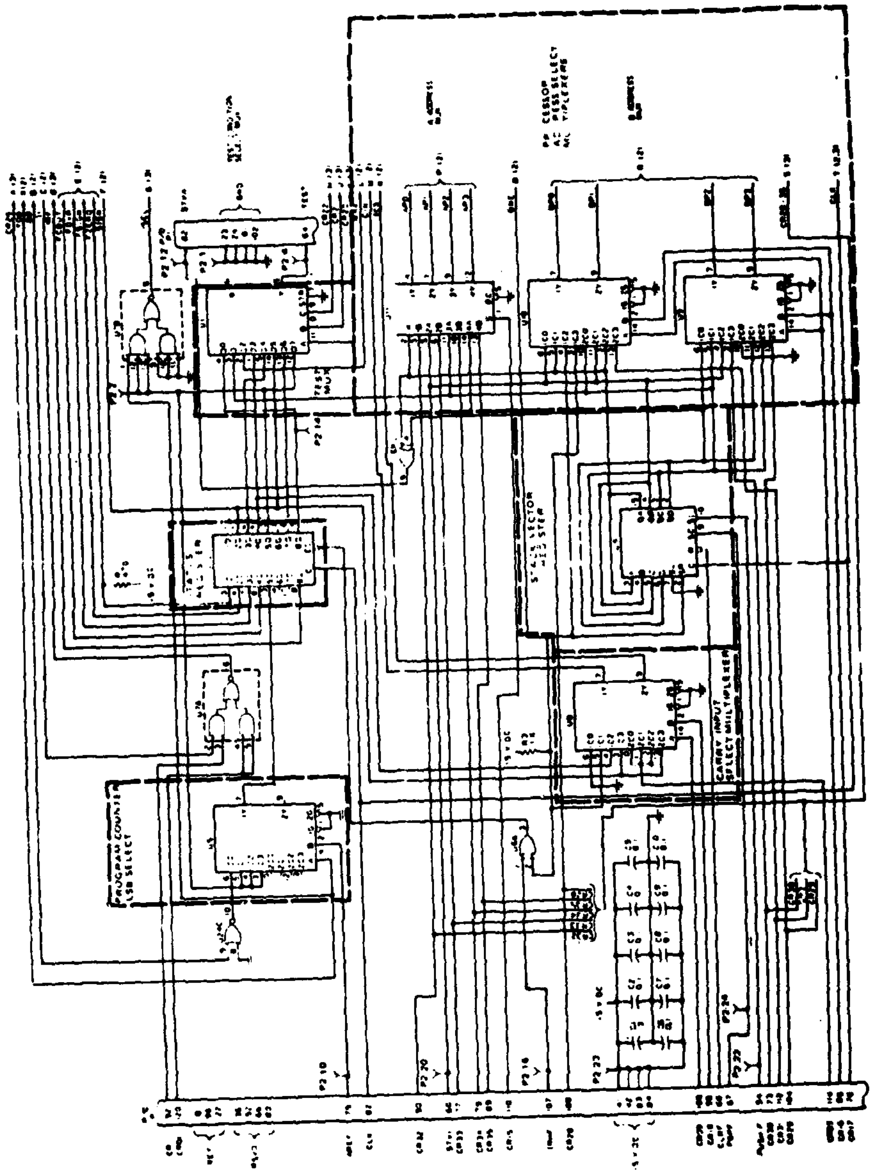


FIGURE C-1. CONTROL CARD SCHEMATIC DIAGRAM (SHEET 3 of 3)



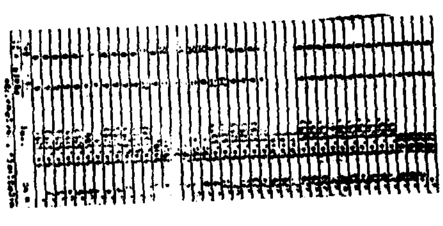


FIGURE C-2. DATA PATH CARD SCHEMATIC DIAGRAM (SHEET 1 of 3)

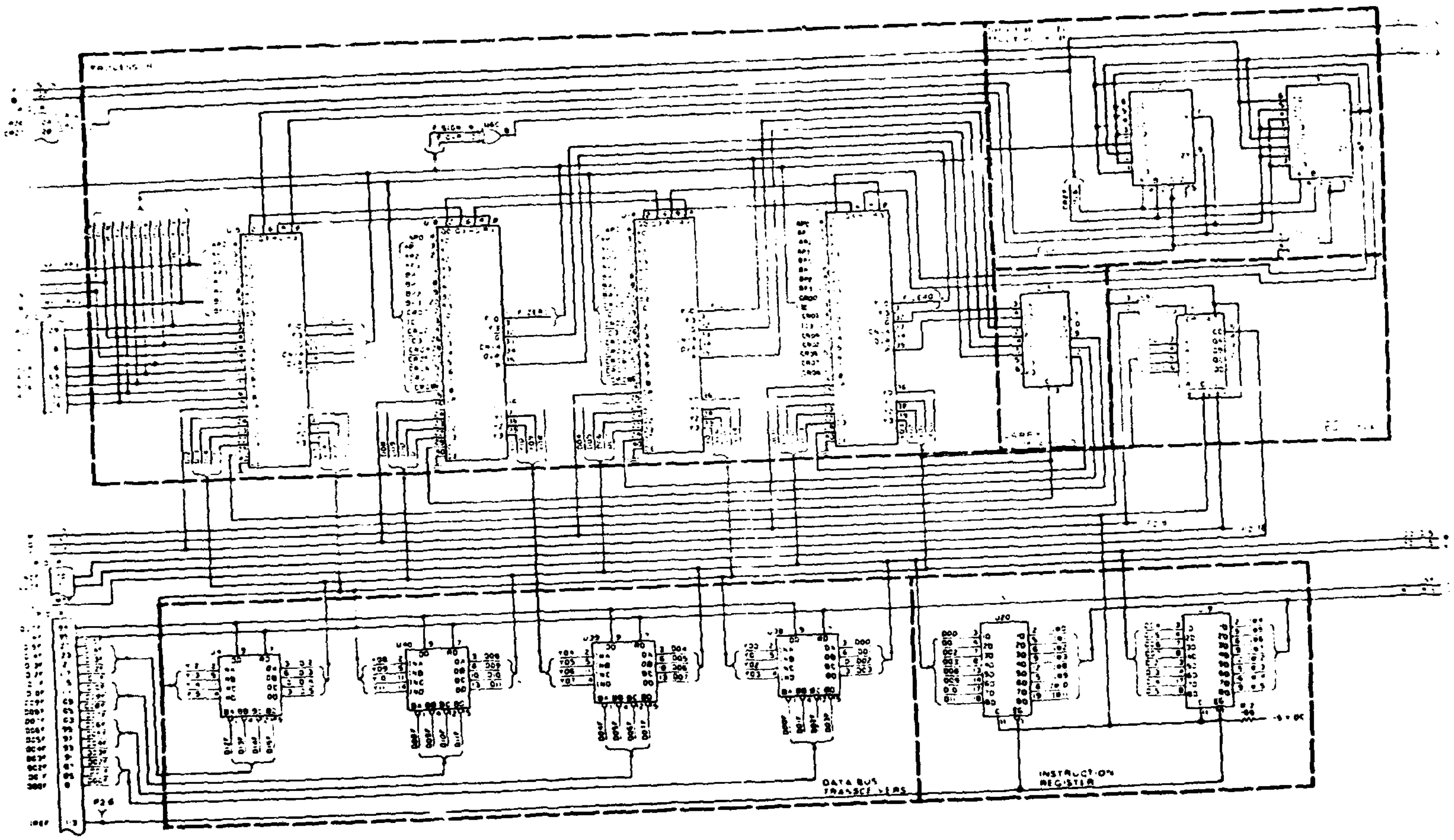


FIGURE C-2. DATA PATH CARD SCHEMATIC DIAGRAM (SHEET 2 of 3)

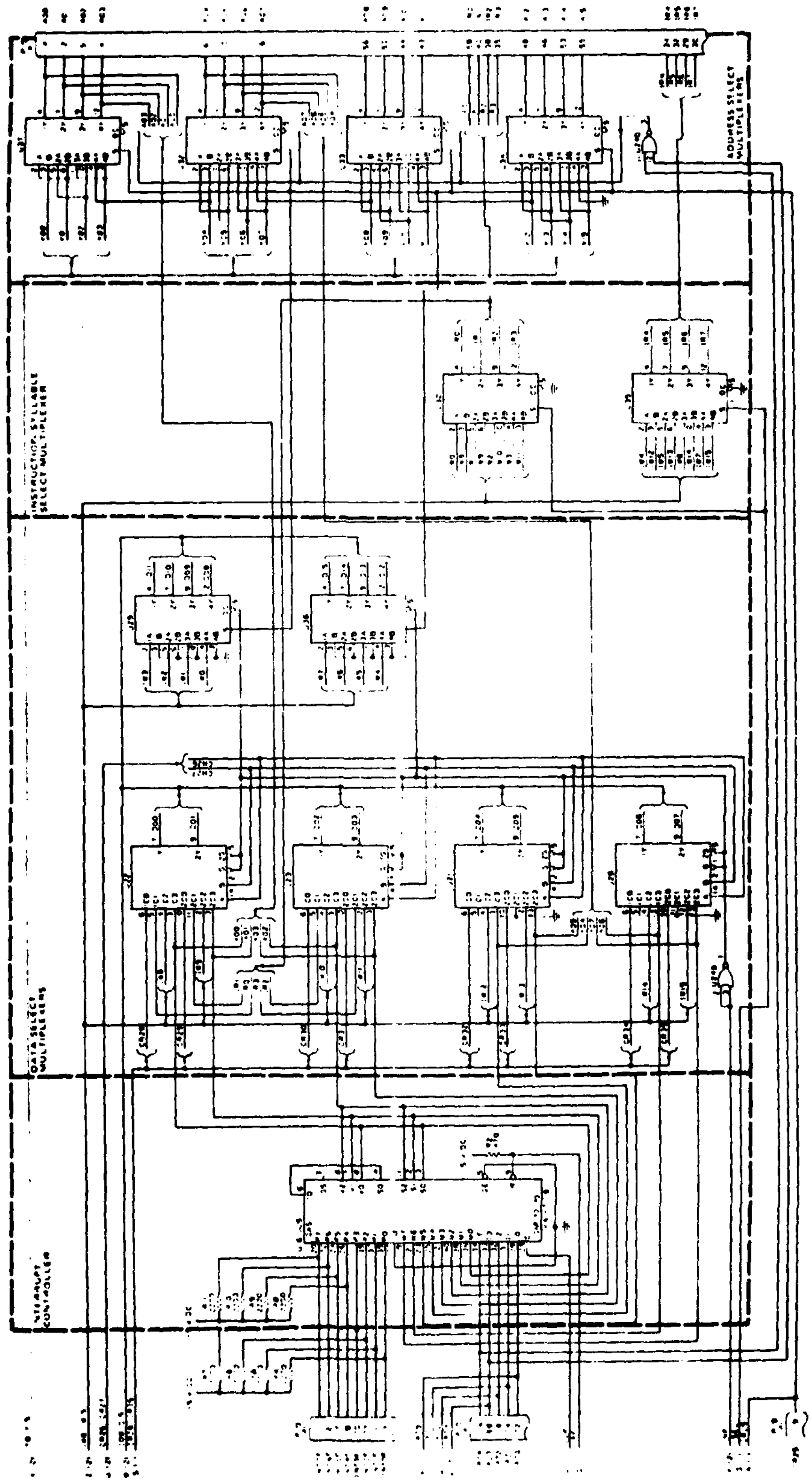


FIGURE C-2. DATA PATH CARD SCHEMATIC DIAGRAM (SHEET 3 of 3)